Sensaphone®

SCADA 3000

User's Manual

Version 2.0

Phonetics, Inc.

Every effort has been made to ensure that the information in this document is complete, accurate and up-to-date. Phonetics, Inc. assumes no responsibility for the results of errors beyond its control. Phonetics, Inc. also cannot guarantee that changes in equipment made by other manufacturers, and referred to in this manual, will not affect the applicability of the information in this manual.

Copyright 1999 by Phonetics, Inc.

Manual Version 2.0 January, 2001

Written and produced by Phonetics, Inc.

Please address comments on this publication to:

Phonetics Inc.

901 Tryens Road

Aston, PA 19014

Sensaphone is a registered trademark of Phonetics, Inc.

All other brands and product names are trademarks or registered trademarks of their respective companies.

Important Safety Instructions

Your SCADA 3000 has been carefully designed to give you years of safe, reliable performance. As with all electrical equipment, however, there are a few basic precautions you should take to avoid hurting yourself or damaging the unit:

Read the installation and operating instructions in this manual carefully. Be sure to save it for future reference.

Read and follow all warning and instruction labels on the product itself.

To protect the SCADA 3000 from overheating, make sure all openings on the unit are not blocked. Do not place on or near a heat source, such as a radiator or heat register.

Do not use your SCADA 3000 near water, or spill liquid of any kind into it.

Be certain that your power source matches the rating listed on the AC power transformer. If you're not sure of the type of power supply to your facility, consult your dealer or local power company.

Do not allow anything to rest on the power cord. Do not locate this product where the cord will be abused by persons walking on it.

Do not overload wall outlets and extension cords, as this can result in the risk of fire or electric shock.

Never push objects of any kind into this product through ventilation holes as they may touch dangerous voltage points or short out parts that could result in a risk of fire or electric shock.

To reduce the risk of electric shock, do not disassemble this product, but return it to Phonetics' Customer Service, or another approved repair facility, when any service or repair work is required. Opening or removing covers may expose you to dangerous voltages or other risks. Incorrect reassembly can cause electric shock when the unit is subsequently used.

If anything happens that indicates that your SCADA 3000 is not working properly or has been damaged, unplug it immediately and follow the procedures in the manual for having it serviced. Return the unit for servicing under the following conditions:

1. The power cord or plug is frayed or damaged.
2. Liquid has been spilled into the product or it has been exposed to water.
3. The unit has been dropped, or the enclosure is damaged.
4. The unit doesn't function normally when you're following the operating instructions.

Avoid using a telephone (other than a cordless type) during an electrical storm. There may be a remote risk of electric shock from lightning.

Do not use the telephone to report a gas leak in the vicinity of the leak.

CAUTION

To reduce the risk of fire or injury to persons, read and follow these instructions:

1. Use only the following type and size batteries: Sealed lead-acid gel cell 12V, 5 to 18AH.

2. Do not dispose of the batteries in a fire. The cell may explode. Check with local codes for possible special disposal instructions.

3. Do not open or mutilate the batteries. Released electrolyte is corrosive and may cause damage to the eyes or skin. It may be toxic if swallowed.

4. Exercise care in handling batteries in order not to short the battery with conducting materials such as rings, bracelets, and keys. The battery or conductor may overheat and cause burns.

FCC Requirements

Part 68: The SCADA 3000 complies with Part 68 of the FCC rules. On the back of the unit there is a label that contains, among other information, the FCC Registration Number and the Ringer Equivalence Number (REN) for this equipment. You must, upon request, provide this information to your local telephone company.

The REN is useful to determine the quantity of devices that you may connect to your telephone line and still have all of those devices ring when your telephone number is called. In most, but not all areas, the sum of the REN's of all devices connected to one line should not exceed five (5.0). To be certain of the number of devices that you may connect to your line, you may want to contact your local telephone company to determine the maximum REN for your calling area.

This equipment may not be used on coin service units provided by the telephone company. Connection to party lines is subject to state tariffs.

Should the SCADA 3000 cause harm to the telephone network, the telephone company may discontinue your service temporarily. If possible, they will notify you in advance. But if advance notice isn't practical, the telephone company may temporarily discontinue service without notice and you will be notified as soon as possible. You will be informed of your right to file a complaint with the FCC. The telephone company may make changes in its facilities, equipment, operations, or procedures where such action is reasonably required in the operation of its business and is not inconsistent with the rules and regulations of the FCC that could affect the proper functioning of your equipment. If they do, you will be notified in advance to give you an opportunity to maintain uninterrupted telephone service.

If you experience trouble with this equipment, or you need information on obtaining service or repairs, please contact:

Phonetics, Inc.

901 Tryens Road, Aston, PA 19014

(610) 558-2700

Fax: (610) 558-0222

The telephone company may ask that you disconnect this equipment from the network until the problem has been corrected or until you are sure that the equipment is not malfunctioning.

Part 15: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Telephone Consumer Protection Act

The FCC Telephone Consumer Protection Act of 1991 makes it unlawful for any person to use a computer or other electronic device, including fax machines, to send a message unless such message contains, in a margin at the top or bottom of each transmitted page or on the first page of the transmission, the date and time it is sent and an identification of the business or other entity, or other individual sending the message, and the telephone number of the sending machine or such business, other entity, or individual. (The telephone number provided may not be a 900 number or any other number for which charges exceed local or long-distance transmission charges.)

To comply with this law, you must enter the following information into your SCADA 3000:

- Date & Time as shown in the System Programming section of this manual.
- Name and telephone number to identify the source of the fax transmission (as shown in the System Programming section of this manual).

General Requirements for all Automatic Dialers:

When programming emergency numbers and (or) making test calls to emergency numbers:

1. Remain on the line and briefly explain to the dispatcher the reason for the call.

2. Perform such activities in the off-peak hours, such as early morning or late evenings.

Canadian Department of Communications Statement

Notice: The Canadian Department of Communications label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective, operational and safety requirements. The Department does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. In some cases, the company's inside wiring associated with a single line individual service may be extended by means of a certified connector assembly (telephone extension cord). The customer should be aware that compliance with the above condition may not prevent degradation of service in some situations.

Repairs to certified equipment should be made by an authorized Canadian maintenance facility designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

CAUTION: Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

The Ringer Equivalence Number (REN) assigned to each terminal device denotes

the percentage of the total load to be connected to a telephone loop which is used by the device to prevent overloading. The termination on a loop may consist of any combination of devices subject only to the requirement that the total of the Ringer Equivalence Numbers of all the devices does not exceed 5.0.

The REN for the SCADA 3000 is 0.7B.

Contents

Important Safety Instructions iii
CAUTION iii
FCC Requirements iv
Telephone Consumer Protection Act v
General Requirements for all Automatic Dialers: v
Canadian Department of Communications Statement v

Chapter 1: INTRODUCTION 1-1
Flexible Communications 1
Built-in Alarm Notification 1
Powerful I/O Expansion Possibilities 1
Built-in Datalogging 1
Powerful Programming Features 1
C-Programming 1
PID Functions 2
GAS FLOW 2
Complete Warranty and Technical Support 2
WHO SHOULD USE THIS MANUAL 2
GENERAL DESCRIPTION 2
About I/Os and Modules 5
Related Publications 5
TECHNICAL SUPPORT 5

Chapter 2: QUICK START GUIDE 2-1
Configuring Communications 1
Create New Unit 2
Communicating with the Unit 2
System Information 3
Dialout Settings 3
Call Reception Settings 4
I/O PROGRAMMING 4

Chapter 3: HARDWARE INSTALLATION 3-1
OPERATING ENVIRONMENT 1
MOUNTING SCADA 3000 1
MOUNTING THE SCADA 3000 POWER SUPPLY (FGD-3100) 3
MOUNTING THE BACKUP BATTERY 3
MODEM AND VOICE BOARDS 4
Modem Card Installation 4
Installation of the Modem Card 5
Voice Board Installation 6

Chapter 4: POWER SUPPLY & BATTERY BACKUP 4-1
POWER REQUIREMENTS 1
GROUNDING AND POWER SURGE PROTECTION 2
POWER SUPPLY FUSE 2
ESTIMATING POWER CONSUMPTION 2

Calculating Power Consumption for the SCADA 3000 2
Power Supply Selection 3
BACKUP BATTERY (OPTIONAL) 4
POWERING THE SCADA 3000 WITH SOLAR CELLS 5
Chapter 5: INPUT WIRING & CONFIGURATION 5-1
General Wiring Considerations 1
TERMINAL BLOCKS 1
INPUT CONFIGURATION 2
Correct Jumper Positioning 2
Normally Open/Normally Closed Dry Contacts 2
1OK Thermistors 3
Analog Voltage Output Transducers (0-5VDC) 4
4-20mA Current Loop Transducers 5
Chapter 6: OUTPUT WIRING 6-1
NOTES ON CORRECT USE 1
Chapter 7: I/O EXPANSION MODULES 7-1
ADDING EXPANSION MODULES 1
Cabling Notes 1
MODULES 1
Power Supply 2
COMMUNICATIONS BUS WIRING 2
Bus Termination 3
Module Types 4
Input Modules 4
Output Modules 4
Chapter 8: SOFTWARE INSTALLATION & SETUP 8-1
MINIMUM REQUIREMENTS 1
INSTALLING THE SOFTWARE 1
Procedure using CD-ROM 2
Procedure using disks 2
UPDATES VIA INTERNET 2
Chapter 9: COMMUNICATING WITH THE SCADA 3000 9-1
Running the Software 1
Communications Programming 1
Cellular Phone Communications 3
Creating New Units 3
Communicating with the Unit 4
System Security 4
User Log-In with Security Enabled 6
Setting Up User Accounts 6
Deleting User Accounts 6
Changing Passwords 6
Radio Vs. Phone Communication 7
RS232 Communication Ports 1 & 2 7
User Inactivity Timer 8

Chapter 10: GENERIC PROGRAMMING 10-1
SAVING AND LOADING PROGRAMMING FILES 1
CONNECTING 1
SYSTEM INFORMATION 3
Unit Name 3
Unit ID Number 3
Slave Address 3
CLOCK 3
Unit Date & Time 3
Auto Daylight Savings 3
DIALOUT SETTINGS 4
Dialing Method 4
Dialing Prefix 4
Maximum Calling Rounds 4
Alpha Pager Speed 4
Voice Repetitions 4
Advanced Button 4
CALL RECEPTION SETTINGS 5
Rings Until Answer 5
Carrier Wait Time 5
Online Timeout 5
Voice Call Security 5
Current User Information 5
UNIVERSAL ADDRESS FORMAT (UAF) 5
Overview 5
How to specify and apply UAFs 6
Input 6
Output 6
Timer 7
Counter 7
Bit 7
Ladder Variable 8
C Variable 8
Gas Flow 8
Alarms 9
Diagnostics 9
Expansion Module Diagnostics 10
I/O PROGRAMMING 11
INPUT TYPE 12
UNITS 12
TABLES AND CALIBRATION 13
UAF 13
OUTPUTS 14
UAF 15
Chapter 11: PROGRAMMING FOR PHONE LINE APPLICATIONS 11-1
DESTINATIONS 1
NAME 1
DESTINATION 1
Numeric Pagers 1
Alphanumeric Pagers 1

E-mail 2
Special Dialing Codes 2
Special Alphanumeric Pager Dialing Codes 3
CALL ZONES 3
ALARM CALL MODE 4
Disabled 4
Until Acknowledged 4
Inform 4
DIAL TYPE 5
INTERCALL DELAY 5
SEND REPORT 5
VOICE PROGRAMMING 6
RECORDING AND PROGRAMMING MESSAGES 6
Items Included in the Status Report 7
Customizing Your Status Report 8
Linking Custom Voice Messages to Data Points 9
Recording and Playback of Messages 10
Local Voice Programming 10
Remote Voice Programming 11
To Hear a Status Report 12
ALARM PROGRAMMING 12
Alarm Name 13
State 13
Status 13
I/O Point Value 13
ALARM SETUP 13
ALARM NOTIFICATION 15
ALARM RECOGNITION 16
Dialout Note: Call Progress 16
ALARM CALL MODE: UNTIL ACKNOWLEDGED VS. INFORM 16
Until Acknowledged 16
Inform 16
Alarm Dialout - Voice 16
Alarm Dialout - Numeric Pager 17
Alarm Dialout - Alphanumeric Pager 17
Alarm Dialout - Modem 17
Alarm Dialout - Fax 17
Alarm Dialout - E-mail 17
ALARM ACKNOWLEDGMENT 17
Alarm Acknowledgment - Voice Dialout 17
Alarm Acknowledgment - Pager Dialout 18
Alarm Acknowledgment - Alphanumeric Pager Dialout 19
Alarm Acknowledgment - Modem Dialout 20
Alarm Acknowledgment - Maximum Calling Rounds 20
ALARM DIALING PRIORITY 20
VIEWING CURRENT ALARMS 20
COMPUTER ALARM RECEPTION 20
Alarm Printing and Logging 21
Printing 22
Logging 22
Audible Alarm 22

REPORTS 22
DATALOGGING 23
DATA LOGGING CAPACITY 24
SELECTING DATA POINTS TO LOG 24
ORDER OF DATA POINTS LOGGED 25
TIME INTERVAL 25
DATA LOGGER START TIME 25
STOPPING THE DATA LOGGER 26
RESETTING THE DATA LOGGER 26
DOWNLOADING THE DATA LOGGER 26
VIEWING THE DATA 26
Graph 27
Export 27
Copy 27
Delete 27
Print 28
New Query 28
Close 28
EVENT LOGGER 28
DOWNLOADING THE EVENT LOGGER 28
VIEWING THE EVENT LOGGER 28
Deleting Events from the Event Log Database 29
Printing, Exporting, and Copying the Event Log 29
POLLING 29
SETTING UP A POLLING SCHEDULE 29
General Set Up 31
CHANGING THE POLLING SCHEDULE 31
INTRODUCTION 1
SELECTING RADIO MODEMS 1
CONFIGURING SCADA 3000 FOR USE WITH RADIO MODEMS 1

Chapter 12: PROGRAMMING FOR RADIO APPLICATIONS 12-1

FLOW CONTROL 2
HOST COMPUTER SETUP 2
COMMUNICATING WITH YOUR SCADA 3000s 6
RADIO MODEM MANUFACTURERS 6

Chapter 13: LCD & KEYPAD PROGRAMMING 13-1

Display Backlight 1
Programming Data Points for Scroll Mode 1
Removing Data Points 2
Grouping Data Points 2
NAVIGATING THE DISPLAY USING THE KEYPAD 2
Scrolling Display 2
USING THE KEYPAD 3
DATA POINTS 3
Inputs 3
Outputs 3
Timers 3
Counters 4

Bits *4*
Ladder Variables *4*
C Program Variable *4*
Gas Flow *4*
Diagnostics *4*
Comm Port Baud Rate 5
Diagnosing Faults 5
ALARMS 5
Acknowledging Alarms from the Keypad *6*
SYSTEM STATUS 6
RETURN 6

Chapter 14: REAL-TIME SCREEN DESIGN 14-1
SUMMARY 1
EDITING 1
Sizing the Palette *1*
Palette Properties *3*
SELECTING SCADA UNITS 3
PLACING OBJECTS ON THE PALETTE 4
Basic Properties *5*
Advanced Properties *5*
Right-click Tools 6
GROUPING COMPONENTS IN A BOX, PANEL OR PAGE 7
Grouping Properties *8*
PLACING A GRAPHIC ELEMENT ON THE PALETTE 10
Placing Datapoint Elements on the Palette *13*
Properties (Label/Value) *14*
PLACING ELECTRICAL ELEMENTS ON THE PALETTE 17
PLACING ROTATING ELEMENTS ON THE PALETTE 19
PLACING CONTAINER COMPONENTS ON THE PALETTE 21
PLACING VALVE COMPONENTS ON THE PALETTE 23
PLACING GAUGES ON THE PALETTE 25
Setting Up Gauges 25
Setting Up the Strip Chart 25
PLACING AUDIO AND VIDEO ELEMENTS ON THE PALETTE 27
Playing an Audio or Video Message 28
PLACING DYNAMIC ELEMENTS ON THE PALETTE 29
REAL-TIME OUTPUT CONTROL 31
SAVING THE REAL-TIME SCREEN 32
LOADING A REAL-TIME SCREEN FOR EDITING 32
RUNNING THE REAL-TIME SCREEN 32

Chapter 15: LADDER LOGIC PROGRAMMING 15-1
RULES/CONVENTIONS OF LADDER LOGIC PROGRAMMING 2
DESCRIPTION OF LADDER ELEMENTS AND FUNCTIONS 2
Logic Definition *2*
XIC Instruction (Examine if Closed) *3*
XIO Instruction (Examine if Open) *3*
OTE Instruction (Output Energize) *4*
OTL Instruction (Output Latch) *4*
OTU (Output Unlatch) *4*

OSR (One-Shot Rising) *5*
TON (Timer On-Delay) *5*
TOF (Timer Off-Delay) *6*
RTO (Retentive Timer) *7*
CTU (Count-Up Counter) *8*
CTD (Count-Down Counter) *9*
RES (RESET) *10*
MOV (Move) *11*
COMPARISON INSTRUCTIONS 11
EQU (EQUAL) *11*
NEQ (NOT EQUAL) *12*
LES (LESS THAN) *12*
LEQ (LESS THAN OR EQUAL) *13*
GRT (GREATER THAN) *13*
GEQ (GREATER THAN OR EQUAL) *14*
MATH INSTRUCTIONS 14
ADD (Addition) *14*
SUB (Subtraction) *15*
MULT (Multiplication) *15*
DIV (Division) *16*
POW (A to the Power of B) *17*
LADDER LOGIC PROGRAM DEVELOPMENT 17
Conserving Memory with Windows 95 and 98 18
STARTING A NEW LADDER PROGRAM 18
Adding Rungs 19
Instruction Toolbars 19
Branching 19
Entering Data Points, Titles, and Comments 20
Changing Instructions 20
Cut, Copy, Paste *21*
Saving and Loading files 21
Deleting Rungs and Instructions 21
Syntax Checking 21
Compiling, Verifying and Uploading 21
Running and Stopping Ladder Execution 22
Ladder Program Statistics 23
Ladder Run Viewer 23
Printing 23
VIEWING AND FORCING LADDER VARIABLES, COUNTERS, TIMERS & BITS 23
Ladder Variables 24
Remote Variable Programming via Touch-Tone telephone *24*
Ladder Timers 24
Ladder Counters 25
Ladder Bits 25
Ladder Program Performance 26
Examples *26*
LADDER PROGRAM EXAMPLES 27

Chapter 16: PROGRAMMING IN C 16-1
Specifications 1
ABOUT THE C LANGUAGE 1
STRUCTURE 2
KEYWORDS 4
Definitions 4
PREDEFINED VARIABLES 6
MONTH, DAY, YEAR, HOURS, MINUTES, & SECONDS 7
FUNCTION LIBRARY 7
ARRAYS 10
SAMPLE PROGRAM 10
ERROR HANDLING 12
C EDITOR 12
Uploading, Compiling, & Starting 13
Downloading 14
Run 14
Start & Stop 15
Synchronous or Asynchronous 15
Control Switch 15
VIEWING THE C VARIABLES 16
Forcing a Value 16
DIFFERENCES BETWEEN STANDARD C AND SCADA 3000 C 16
C PROGRAMMING EXAMPLES 18

Chapter 17: PID PROGRAMMING 17-1
PID ALGORITHM 2
TUNING 3

Chapter 18: GAS FLOW MEASUREMENT 18-1
ORIFICE PLATE 1
POSITIVE DISPLACEMENT 1
Definitions 4
CALCULATED INFORMATION 5
Hourly Data (updated on the hour) 5
Daily Data 5
Monthly 6

Chapter 19: SYSTEM DIAGNOSTICS 19-1
MAIN BOARD DIAGNOSTICS 1
MODULE COMMUNICATIONS DIAGNOSTICS 2

Chapter 20: INTERNET WEB STATUS 20-1
REQUIREMENTS 1
INTERNET SETTINGS 1
GLOBAL WEB STATUS SETTINGS 2
SELECTING DATA POINTS FOR WEB STATUS POLLING 4
FORMATTING THE WEB PAGE 5
Browser Refresh 5
Settings for Top Left Corner Logo 5
Settings for Bottom Logo 5

WEB STATUS OPTIONS 6
Alarm Options 6
Online Options 6
Local Web Pages & Web Page Filenames 6
FTP Delivery Options 6
Instant Web Page 7
SETTING UP A POLLING SCHEDULE FOR WEB STATUS 7
FREQUENTLY ASKED QUESTIONS 7

Chapter 21: EMAIL SERVER OPTION 21-1
REQUIREMENTS 1
HOW DOES IT WORK? 1
PROGRAMMING 1

Chapter 22: PC ALARM DIALOUT 22-1
Computer Requirements 1
SOFTWARE SETUP 1
Modem Commands and Modem Messages 2
Email Settings 2
Telephone Voice Volume Setting 3
SCADA 3000 PROGRAMMING 3
Voice Messages 4
PROGRAMMING DESTINATIONS 5
ALARM PROGRAMMING 6
Alarm Acknowledgement 6
SAMPLE ALARM DIALOUT SEQUENCE 6
Voice 6
Numeric Pager 7
Alphanumeric Pager 7
FAX and E-mail 7

Chapter 23: DISTRIBUTED CONTROL USING VISUAL BASIC SCRIPT 23-1
REQUIREMENTS 1
DISTRIBUTED CONTROL SCRIPT EDITOR 1
REFERENCING SCADA 3000 I/O POINTS 3
Sample Application: 3
Syntax Checking 4
Saving your Program 5
Running your Program 5
PROPERTIES OF THE SCRIPT COMPONENT 5
ABOUT THE VISUAL BASIC LANGUAGE 6
STRUCTURE 6
KEYWORDS 7
DEFINITIONS 7
Dim Statement 7
If...Then...Else Statement 7
For...Next Statement 8
Do...Loop Statement 8
OPERATORS 9
PROGRAMMING METHODOLOGY 10
SAMPLE PROGRAM 11

Appendix A: CHECKING YOUR SCADA 3000 FOR PROPER OPERATION..... A-1
Appendix B: SCADA 3000 SPECIFICATIONS B-1
Appendix C: LIST OF SYSTEM EVENTS..... C-1
Appendix D: 10k THERMISTOR CURVE DATA..... D-1
Appendix E: COMMUNICATIONS PORTS E-1
Appendix F: UPDATING SCADA 3000 FIRMWARE F-1
Appendix G: SOFTWARE INTERNET UPGRADE G-1
STARTING THE PROGRAM 1
Verify 1
Retrieve 1
Install 1
OPTIONS 2
Local Tab 2
File Tab 3
Proxy Tab 3
Server Tab 4
Appendix H: TROUBLESHOOTING H-1
Appendix I: SCADA 3000 ACCESSORIES I-1
Appendix J: SCADA 3000 REPLACEMENT PARTS J-1
Appendix K: RETURNING THE UNIT FOR REPAIR K-1
IMPORTANT INFORMATION FOR CANADIAN CUSTOMERS 2
Appendix L: PROGRAMMING FORMS L-1
Index
Warranty

Chapter 1: INTRODUCTION

Congratulations on your purchase of the Sensaphone SCADA 3000, the intelligent RTU and software system for powerful, low cost SCADA solutions. The SCADA 3000 comes complete with all the software you'll need, in a single, integrated package. Your entire system is easier to set up, easier to program and support, and much more cost-effective than other traditional SCADA systems.

Flexible Communications

SCADA 3000 RTUs communicate with the host PC by standard data radio, ordinary phone lines, or a combination of both. With an adaptor, even cellular phone communication is possible.

Built-in Alarm Notification

The Sensaphone SCADA 3000's complete set of communications options and capabilities can save you expensive delays and downtimes. Whether you use phone lines or radio, the SCADA 3000 system can report alarms using custom voice, numeric pagers, alphanumeric pagers, fax and even e-mail.

Powerful I/O Expansion Possibilities

Standard on each SCADA 3000 are 16 universal inputs and 8 relay outputs. The universal inputs can accept contact closures, 4-20mA transducers, 0-5Volt sensors, and even thermistors. As your enterprise grows and you bring new machinery or facilities on-line, the Sensaphone SCADA 3000 can expand to meet your needs. A variety of expansion modules are available to add specialized inputs such as thermocouples or high speed pulse counters, as well as additional universal inputs. Outputs can be expanded to add more relay channels, or 4-20mA analog output channels. You can add modules to the SCADA 3000 and expand up to a total of 144 I/O points.

Built-in Datalogging

Fully programmable time-stamped data comes standard with SCADA 3000, which not only internally logs and stores data, but produces reports via phone modem *directly* to your main computer or even a fax machine.

Powerful Programming Features

Ladder Logic, the most easily understood and universal control language in use today is made even easier to use in the SCADA 3000. Standard ladder instructions are built in along with counters, timers, and even analog capability. The free software package comes complete with a Windows®-based drag-and-drop ladder editor to significantly reduce your set-up time.

C-Programming

The Sensaphone SCADA 3000 has built-in C-programming to handle any control algorithms, which may be too complex for Ladder Logic. Though the C-program can run independently of ladder programs, we have built in the capability of both programs to share and pass information back and forth for ultimate control. And, of course, the necessary software to develop C-programs is included in our free software package!

PID Functions

Control needs are rounded out with the SCADA 3000's built-in Proportional Integral Derivative control functions. Any analog input and output channel pair can be associated with up to 8 PID algorithms. Each algorithm has its own set of fill-in-the-blank parameters.

Gas Flow

The SCADA 3000 has the capability of measuring natural gas flow. Calculations for up to four meter runs can be performed simultaneously. The SCADA 3000 supports two methods of gas flow measurement: orifice plate and positive displacement (turbine meter). The formulas used for gas flow calculations are in accordance with A.G.A. Report No. 3, A.G.A. Report No. 7, and A.G.A. Report No. 8. In addition to real-time measurement, the SCADA 3000 also computes hourly, daily, and monthly data.

Complete Warranty and Technical Support

All Sensaphone products are manufactured entirely in the United States. Phonetics stands behind the SCADA 3000 with a 3-year comprehensive warranty and technical support. Support is available at (610)558-2700, or through our website at www.sensaphone.com, or by e-mail at support@sensaphone.com.

Who should use this manual

The very nature of this product requires that the user have some knowledge of electrical wiring and computer-based control. This manual is written assuming that the user has this type of background. Safety issues must be considered in the design of any control system to prevent injury and also to insure that it will meet Local and National Electric Codes. If you intend to connect high voltage circuits to the unit and you do not have experience working with high voltages, Phonetics strongly recommends that you obtain this knowledge or have the work performed by a qualified individual.

General Description

Pictured below is the SCADA 3000 main unit (often referred to in this manual as the CPU or motherboard). Descriptions of the numbered elements follow.

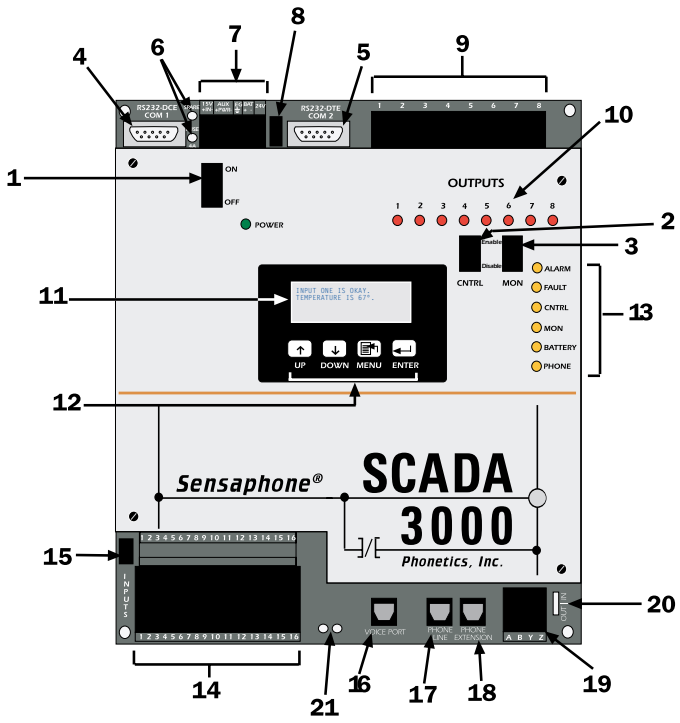


Figure 1: SCADA 3000 CPU

1. Power Switch: Switches power from the "15V IN" terminals to the SCADA 3000 circuitry. Also applies power to the "AUX PWR" terminals and will charge and/or operate from a battery connected to the "BAT" terminals.

2. Control Enable/Disable Switch: Manual override switch to disable or stop the ladder, C control, and PID programs from running. The software can be used to start and stop the control programs only if the switch is in the Enable position. If the switch is set to the Disable position the programs cannot be started.

3. Monitor Enable/Disable Switch: Manual override switch to disable or stop the unit from monitoring for alarms, thereby preventing alarm dialout. The software can be used to enable and disable monitoring only if this switch is in the Enable position. If the switch is set to the Disable position, then monitoring cannot be enabled via software control.

4. RS232 DCE Port (COM1): Programming port for connection to an IBM or compatible computer. Defaults to 38.4 Kbps. (See Appendix D for details)

5. RS232 DTE Port (COM 2): Multi-use communications port for use with a radio modem, or other optional device. Defaults to 38.4 Kbps. (See Appendix D for details)

6. MAIN POWER FUSE & 24V SUPPLY FUSE: 4A 250VAC main power fuse and one 0.5A 250VAC 24V supply fuse.

7. POWER TERMINAL STRIP

15V IN: Main power input.
AUX PWR: Output power for I/O expansion modules.
EG: Earth ground.
BAT: External 12V battery input.
24V OUT: 24V power source for powering accessories or transducers.

8. EG Header: Jumper to connect/disconnect circuit board ground to earth ground.

9. OUTPUT Terminal Strip: Eight latching relay outputs rated to 125VAC 2A.

10. OUTPUT LEDs: LEDs to indicate the ON/OFF status of each output relay (0-7).

11. LCD Display: 80 character back-lit programmable display for showing data points, system status, and alarms.

12. Keypad Buttons: Four keypad buttons that work in combination with the display to access information and program the display.

13. Status LEDs

ALARM: When flashing, this LED indicates that an unacknowledged alarm exists. When on steady it indicates that an alarm exists but has been acknowledged.

FAULT: This LED indicates a problem with the SCADA 3000 or one of the I/O modules. The problem can be identified by viewing the system diagnostics and unit event log.

CNTRL (Control Program Enabled): Indicates that the control program is enabled.

MON (Monitoring Enabled): Indicates that alarm monitoring is enabled. If an alarm occurs the SCADA 3000 will initiate its alarm dialout sequence.

BATTERY OK: When on steadily this LED indicates that the external battery is fully charged. When blinking it indicates that the battery is low. When off it indicates that the battery is either not connected or very low.

PHONE Status: This LED indicates if the SCADA 3000 is currently using the phone line.

14. Input Terminal Strip: These terminals are for connecting input sensors and transducers. The upper row is the positive input and the lower row is the negative input. The inputs are differential for certain jumper configurations so be sure to wire them in direct vertical pairs (eg. The positive input wire directly above the negative input wire).

15. Input Configuration Jumpers: These jumpers are used to condition the input for different types of sensors. Use needlenose pliers to move them to the appropriate position. (See Chapter 5: Input Wiring)

16. Voice Port: This jack is for plugging in a telephone to listen to a status report or record messages if the voice option is installed.

17. Phone Line: This is the jack for connecting to the telephone network.

18. Phone Extension: This jack is for connecting telephone extensions to the same line as the SCADA 3000. This jack features Line Seizure so that if the SCADA 3000 needs to make a call it will disconnect the extensions and seize the line for its own use.

19. Expansion Bus Terminal Strip: This terminal strip is for connecting the 4-wire communications link to the I/O expansion modules.

20. Bus Termination: This jumper selects whether the expansion bus is terminated or not.

21. Spare Fuses: Spare fuses are provided for the main power and 24V supply.

About I/Os and Modules

The SCADA 3000 comes with 16 universal inputs and 8 relay outputs. Additional inputs and outputs may be added by connecting additional I/O modules to the main unit. Up to 15 modules may be connected to the unit providing as many as 144 total I/O points. The modules connect to the unit via a 4-wire communications bus. They may be located up to 2000 feet away from the main unit. Modules are available to connect a variety of I/O devices:

FGD-3010 Scada 3000 8 Channel Universal Input Module
FGD-3020 Scada 3000 4 Channel Pulse Count Input Module
FGD-3030 Scada 3000 4 Channel Thermocouple Input Module
FGD-3040 Scada 3000 8 Channel Relay Output Module
FGD-3050 Scada 3000 4 Channel Analog Output Module
FGD-3060 Scada 3000 8 Channel Annunciator Module

Related Publications

Ladder logic programming and C-programming are widely used languages. This manual does its best to present these topics as they apply to the SCADA 3000; however, there are numerous texts available which provide extensive information specific to ladder & C-programming.

Each I/O module comes with its own documentation for insertion into this binder. For your convenience, the documentation for each separate module includes a labeled divider to provide fast & easy access to that section.

For the latest technical information check our website at www.sensaphone.com

Technical Support

If any questions arise upon installation or operation of the SCADA 3000, please contact Phonetics Technical Service Department at the number shown below and have the following information:

- Date of Purchase _____
- Serial Number _____ (bottom of unit)
- Firmware version_____/_____ (while on-line, click Diagnostics>Processors)
  COMM CONTROL
- PC Software Version _____(while running the software, click Help>About)

Technical Support is available from 8:00 AM to 5:00 PM, EST. You may also e-mail your questions to support@sensaphone.com

Phonetics, Inc.
901 Tryens Road
Aston, PA 19014
Phone: (610)558-2700
Fax: (610)558-0222
www.sensaphone.com

Chapter 2: QUICK START GUIDE

This section presents a useful shorthand guide to configuring your SCADA 3000 unit for the first time. More detailed information on the processes covered can be found in Chapters 8, 9, and 10 of this manual.

Install the software from your CD-ROM or other media.

Double click on the SCADA 3000 icon on your desktop. The Contr ol Center for m will appear .

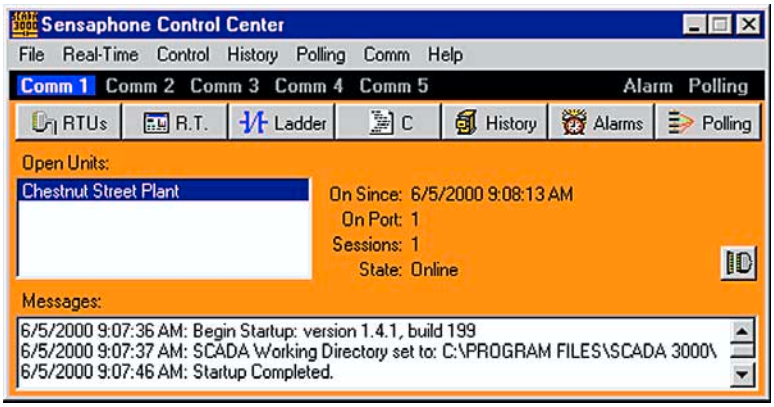

Sensaphone Control Center
File Real-Time Control History Polling Comm Help
Comm 1 Comm 2 Comm 3 Comm 4 Comm 5 Alarm Polling
RTUs R.T. Ladder C History Alarms Polling
Open Units:
Chestnut Street Plant
On Since: 6/5/2000 9:08:13 AM
On Port: 1
Sessions: 1
State: Online
Messages:
6/5/2000 9:07:36 AM: Begin Startup: version 1.4.1, build 199
6/5/2000 9:07:37 AM: SCADA Working Directory set to: C:\PROGRAM FILES\SCADA 3000\
6/5/2000 9:07:46 AM: Startup Completed.

Figure 1: Control Center form

The Control Center form is the main menu for accessing all features of the SCADA 3000. Connect the SCADA 3000 COM 1 port to a serial port on your computer to communicate directly or connect the SCADA 3000 to a telephone line to communicate via modem.

Configuring Communications

1. Choose the **Comm** pulldown menu.
2. Select “Setup.”

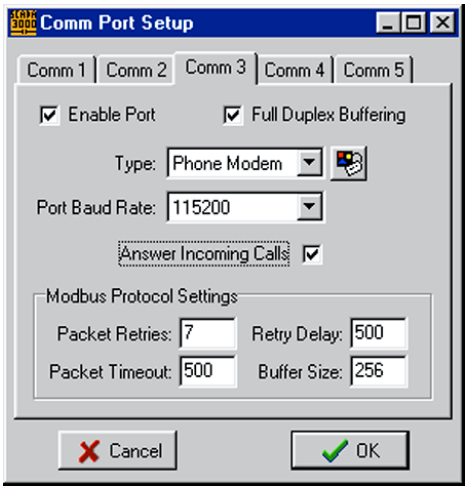


Figure 2. Communications Port Setup form

3. Configure Communications port(s).

a. Click to check "Enable Port" for each Comm Port you are setting up.

b. Select the Type of communication you will be using to communicate with the SCADA 3000--Serial Port, Phone Modem or Radio Modem.

c. Select appropriate baud rate for the modem or SCADA 3000 connected to the port. The SCADA 3000 DCE serial port defaults to 38400 bps.

d. Click OK when you've finished configuring the ports, to close this form.

Create New Unit

1.From Contol Center form: Click on the RTUs button or select File>RTUs. The Unit Database form appears.

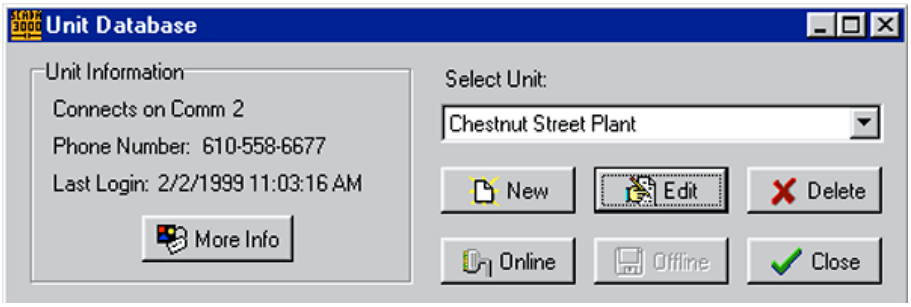


Figure 3. Unit Database form

2. Click on "New."

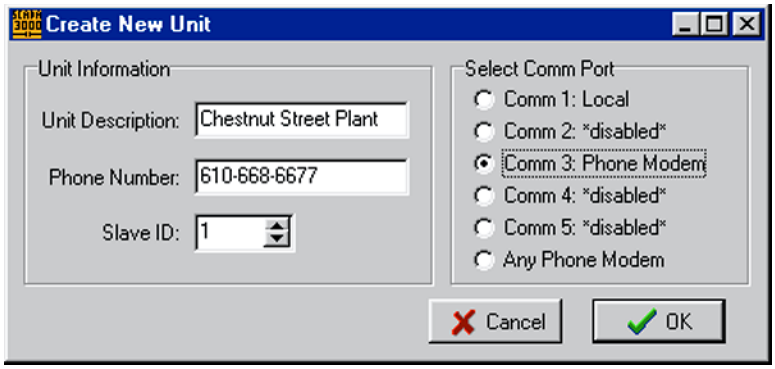


Figure 4. New Unit form

3. Type in the name you want to assign the new unit.

4.Type in the phone number or, if on a radio network, set the Slave ID.

> **Note**: If you are on a Phone communication network and have only one unit per Comm port, the slave IDs can all be set to "1" without conflict. However, if you are using a Radio Communication network, each unit *must* have a unique slave ID.

5. Select the Comm port to which the unit is connected.

6. Click OK.

Communicating with the Unit

Return to the Unit Database form and click the Online button.

An interim Connecting form appears briefly as your computer connects to the unit.

The Main Programming form appears. You may now begin programming.

The Register Verification process will run in the background until it finishes comparing the software in the computer with that in the RTU. This procedure is required to ensure data integrity among multiple users of the unit. The Register Verification status bar shows the download progress and can be hidden or displayed at the user's discretion by clicking on *View>Register Verification* from the main menu and selecting *Show Register Verification*. If you choose not to view Register Verification, the clock showing Local Unit Time indicates if you are communicating with the RTU. If the time on the clock is not moving (advancing), then you are not in fact connected. If you view Register Verification at that point, you will not see any progress in the status bar.

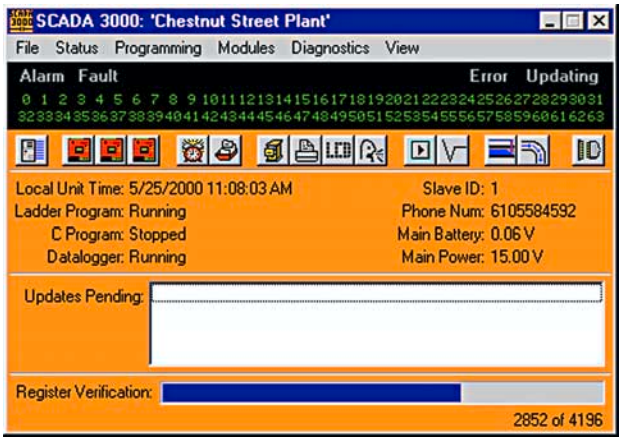


Figure 5. Main Programming form

Select the Programming pull-down menu

Choose "System" to bring up the system form.

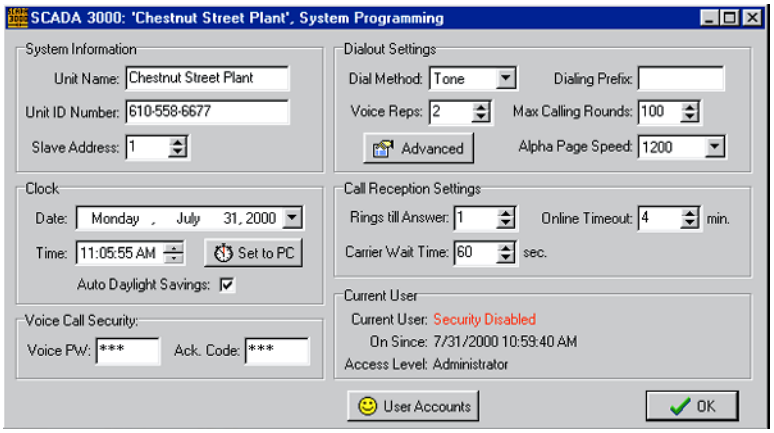


Figure 6: System form

System Information

Both the Unit Name and ID Number should be the same as those you assigned on the New Unit form.

Dialout Settings

1. Dial Method: Can be either tone or pulse. Default setting is Tone.

2. Dialing Prefix: for E-Mail prefixes only. If you need to dial a "9" to get an outside line before dialing an email server number, enter that here. For other dialout prefixes see the "Destinations" section of the manual.

3. Voice repetitions: The number of times the SCADA 3000 repeats alarm messages per call. Default setting is 2.

4. Maximum calling rounds: the number of times a SCADA 3000 will dial through your list of destinations to deliver its messages. Default is 100.

5. Alphanumeric Pager Speed: Default setting is 1200 Baud

Call Reception Settings

1. Rings till Answer: the number of times the SCADA will let the phone ring before answering an incoming call. Default is 1 ring; range is 1-20 rings.

2. Carrier Wait Time: The amount of time SCADA will wait for a modem connection when receiving an incoming call. Default is 60 seconds; range is 0 to 90 seconds.

> **Note**: Don't set this time too short, or a modem connection may never be established.

3. Online Timeout: Automatically logs you off if there are no communication requests from the computer within the specified time period. The default setting is 4 minutes. Range: 0-255, with "0" representing a disabled setting.

Click OK.

I/O PROGRAMMING

Click on the first of the three Motherboard and Module I/O programming buttons to bring up the I/O form for the first 8 inputs, the second button for the next 8 inputs and the third button for the 8 outputs.

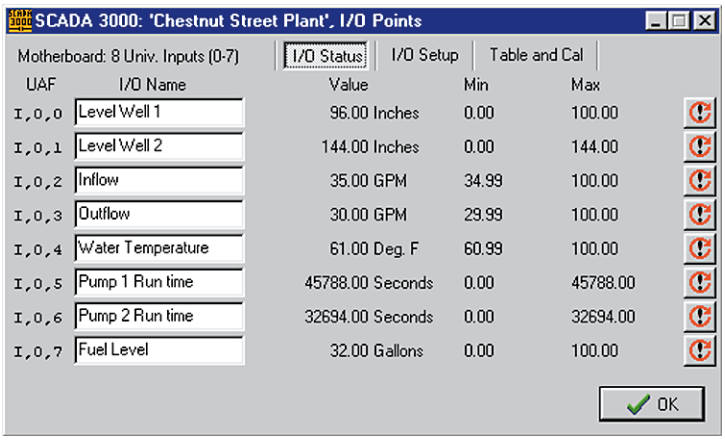

SCADA 3000: 'Chestnut Street Plant', I/O Points
Motherboard: 8 Univ. Inputs (0-7)
I/O Status
I/O Setup
Table and Cal
UAF
I/O Name
Value
Min
Max
I,0,0 Level Well 1 96.00 Inches 0.00 100.00
I,0,1 Level Well 2 144.00 Inches 0.00 144.00
I,0,2 Inflow 35.00 GPM 34.99 100.00
I,0,3 Outflow 30.00 GPM 29.99 100.00
I,0,4 Water Temperature 61.00 Deg. F 60.99 100.00
I,0,5 Pump 1 Run time 45788.00 Seconds 0.00 45788.00
I,0,6 Pump 2 Run time 32694.00 Seconds 0.00 32694.00
I,0,7 Fuel Level 32.00 Gallons 0.00 100.00
OK

Figure 7: I/O Points

The form above shows the first eight inputs on the main unit, 0-7, retrieved by clicking on the first of the three buttons.

Click the "I/O Setup" button to open the setup form. Here you assign each input a name, a type, and the units of measurement applying to that type.

To set the Low and High values for inputs, and the calibration factor if any, click on the "Table and Cal" button.

Click on OK to close the form once you've programmed your inputs.

Chapter 3: HARDWARE INSTALLATION

This chapter provides the information necessary to install the Sensaphone SCADA 3000. Correctly installing the unit will ensure proper functioning and maximum service life. Please read the entire chapter before attempting installation.

OPERATING ENVIRONMENT

SCADA 3000 should be mounted and operated in a clean, dry and safe environment. Do not mount the unit where it will be subject to shock and vibration. Do not mount the unit where it will be subject to dirt, dust or moisture. Ideally the unit would be mounted in a steel or a fiberglass NEMA-4 enclosure. **Do not mount the unit or the expansion modules close to motor starters, contactors or relays that switch inductive loads.** These devices generate large electromagnetic fields that can cause the SCADA 3000 to malfunction. Where this is unavoidable, mount the unit and modules in a separate, grounded, steel enclosure. This will shield the unit from harmful electrical interference.

The temperature range the SCADA 3000 can operate in is 32°F to 158°F (0°C to 70°C). If you require SCADA 3000 to operate in a below freezing environment, you must take safe and practical measures to keep the unit's temperature above 32°F or it will not operate reliably.

CAUTION: SCADA 3000 is a sensitive electronic device. Personnel and work area should be grounded before handling this device. Do not install SCADA 3000 near any strong electrostatic, electromagnetic, magnetic or radioactive fields. Do not expose it to fumes or corrosive vapors.

MOUNTING SCADA 3000

When you receive SCADA 3000, carefully remove it from the box. On the top and bottom of the enclosure are mounting holes to attach the unit to either a panel or wall. The mounting surface should be sturdy enough to support 15 lbs. The unit should be mounted using four #10-32 bolts where appropriate, or four #10 tapping screws. (The screw kit for the SCADA 3000 includes 4 #10-32 screws, 4 #10-32 nuts, 4 #10 lockwashers, and 2 #6-32 screws, 2 #6-32 nuts, and 2 #6 lockwashers. The #6 hardware is for the power supply.) When mounting the unit to a wall make sure the mounting screws fully engage a solid member (for example, a stud) of the support structure. Mount SCADA 3000 in an upright position so that you can easily read the display and connect wires to the terminal strips. There must be a power outlet and, if you are using telephone communications, a telephone jack nearby. The dimensions of the full enclosure are: 12.2" x 9.4" x 2.0". See Figure 1 on the following page.

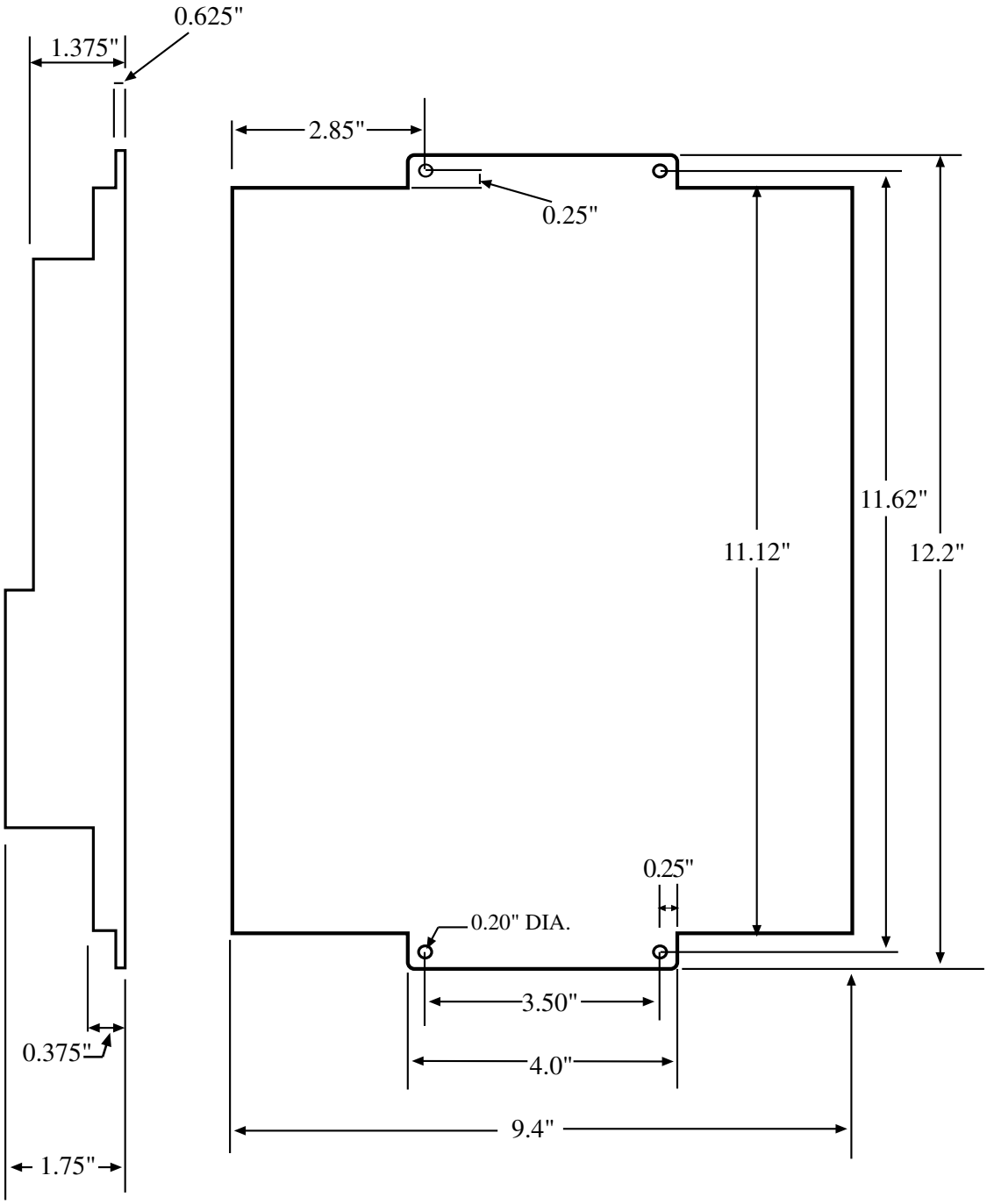


Figure 1: Mounting Dimensions

MOUNTING THE SCADA 3000 POWER SUPPLY (FGD-3100)

The Power Supply for the SCADA 3000 may be mounted either flat or sideways, utilizing the mounting holes provided. Mounting hardware for the power supply (#6-32 screws, nuts and washers) has been included in the nuts and bolts hardware package. See the Figure below for dimensions.

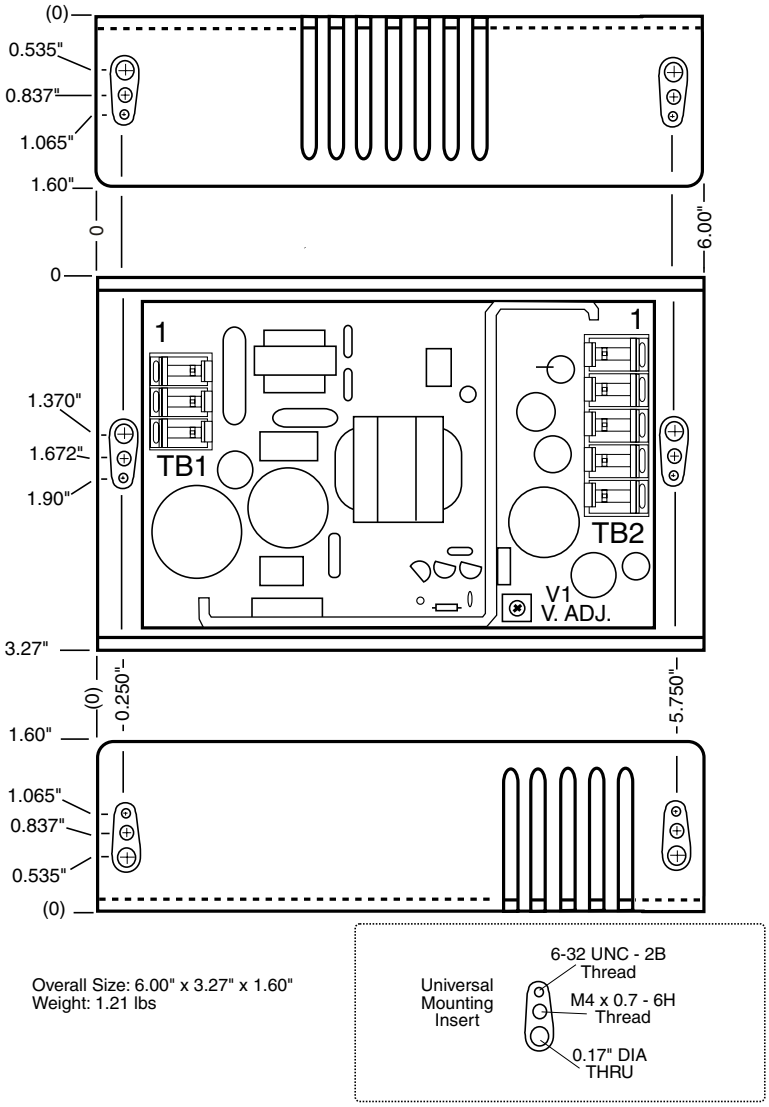


Figure 2: Power Supply Mounting Dimensions, side and front mount

MOUNTING THE BACKUP BATTERY

You may use either a 5.2 AH battery (FGD-3200) or 18.0 AH battery (FGD-3210) with the SCADA 3000. When a battery is ordered, it will arrive with a screw kit (3 #10-32 screws, 3 nuts, and 3 lockwashers) and appropriate battery bracket. The bracket is designed to be mounted with the restraining bar either on top or on the bottom of the battery, depending upon mounting environment. For bracket mounting dimensions, see Figures 3 and 4.

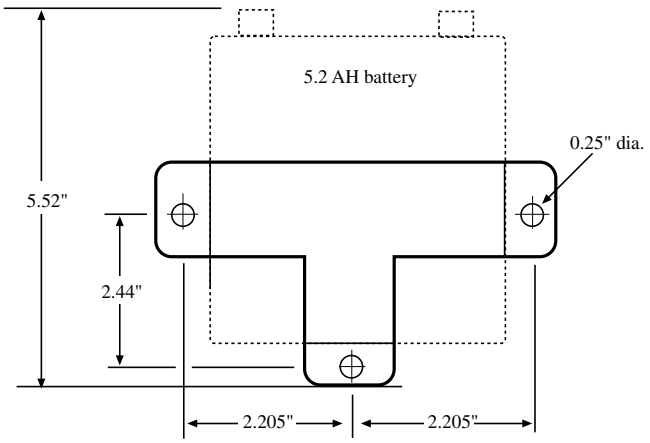


Figure 3: Bracket for 5.2 AH battery (FGD-3200)

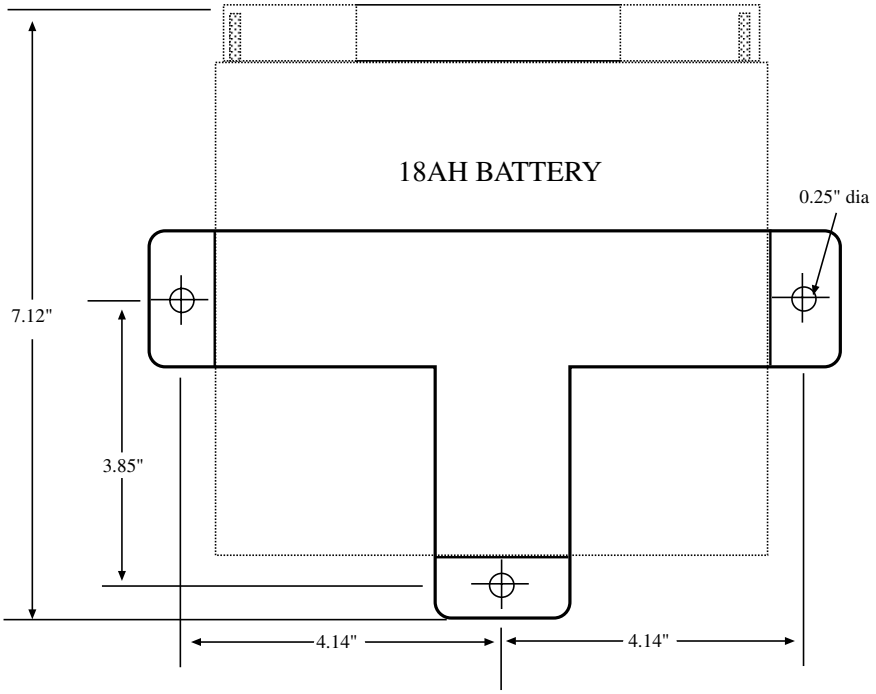


Figure 4: Bracket for 18AH battery (FGD-3210)

MODEM AND VOICE BOARDS

Both the FGD-3400 Modem card and the FGD-3500 Voice Board are optional components of the Sensaphone SCADA 3000. Most people will choose to have these features factory-installed when they purchase their unit. However, if you determine after purchasing that you wish to add either Voice or Modem function, we include instructions on installing both. You can also use these instructions to remove the modem card or voice board for servicing.

Modem Card Installation

The modem card is required if you want your SCADA 3000 unit to provide real-time input/output information via phone line, send fax alarms and reports, contact alphanumeric pagers, or

send E-mail. The modem card utilizes internationally accepted standard modulation methods and protocols. It features the ease of Plug and Play installation: The SCADA 3000 operating system *will automatically* detect its presence once installed.

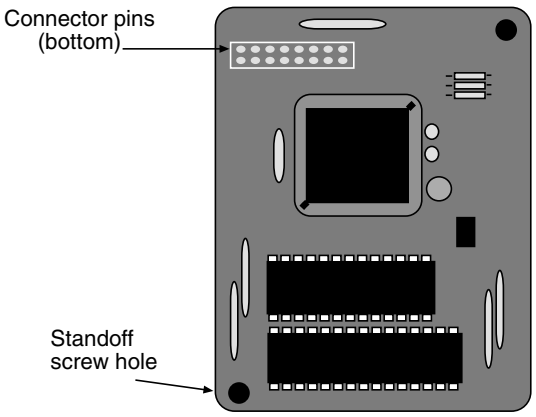


Figure 5: Modem card

The FGD-3400 modem card offers:

Data modem throughput up to 115.2 kbps

- 33.6 kbps and V.34
- V.32 bis, V.32, V.22 bis, V.22A/B, V.23, and V.21; Bell 212A and 103
- V.42 LAPM, MNP 2-4, and MNP 10 error correction
- V.42 bis and MNP 5 data compression
- MNP 10EC™ enhanced cellular performance
- Error correction and data compression (ECC) performed in the modem using 32k bytes of external RAM to increase data throughput typically by a factor of four. Fax modem Group 3 send rates up to 14400 bps
- V.33, V.17, V.29, V.27 ter, and V.21 channel 2
- Supports T.30 protocol

Installation of the Modem Card

1. Turn off the SCADA 3000 unit.

2.While wearing a grounding wrist strap, remove the four screws on the top cover and slowly remove it, being careful to manually unplug the ribbon connection from the keypad in the center.

Note: Pulling the cover off quickly can damage the LCD ribbon or connector.

3. The modem card connector is below and to the right of the LCD window. There are two standoff screw mounting holes in the lower righthand quadrant of the main board, one for the upper right corner of the card, and one for the lower left corner of the card. (See Figure 6)

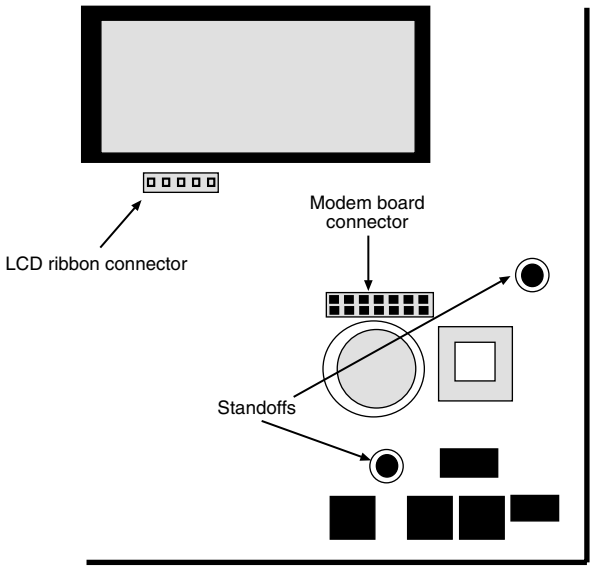


Figure 6: Modem mounting on the main board

Line the card up with the connector and push it carefully but firmly into place. With the two screws provided with the modem card, secure it to the two standoffs.

4. Replace the cover, manually reconnecting the keypad ribbon to the board. Replace the four screws. Your modem card is now installed and ready for use.

Voice Board Installation

The FGD-3500 voice board provides high-quality, long-duration record/playback messaging for your SCADA 3000 system. It features CMOS devices for low-power operation, automatic gain control, anti-aliasing filters, smoothing filters and high-density storage arrays. Recordings are stored in on-chip non-volatile memory cells, providing zero-power message storage. Voice and audio signals are stored *directly* into memory in their natural form, providing high-quality, solid-state voice reproduction.

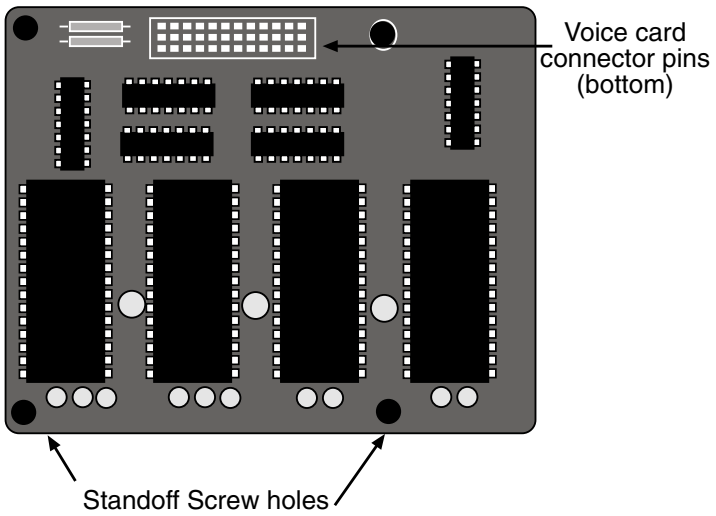


Figure 7: Voice Board

The FGD-3500 features:

- 270 seconds of custom message storage
- 90 seconds of pre-recorded vocabulary
- High-quality natural voice reproduction
- Low power design
- Zero-power (non-volatile) message storage
- 100,000 record cycles per message
- Message sampling rate: 5.3KHz
- Message bandwidth: 2.3KHz

Installation

1. Turn off the SCADA 3000 unit.

2. For safety, put on a grounding wrist strap. Remove the four screws on the top cover and slowly remove it, being careful to manually unplug the ribbon connection from the keypad in the center.

Note: Pulling the cover off quickly can damage the ribbon or connector.

3. The 36-pin voice board connector socket is below and to the left of the LCD window. There are three standoff screw mounting holes in the lower left quadrant of the main board—one for the upper left corner of the voice board, and two for the bottom corners. (See Figure 8)

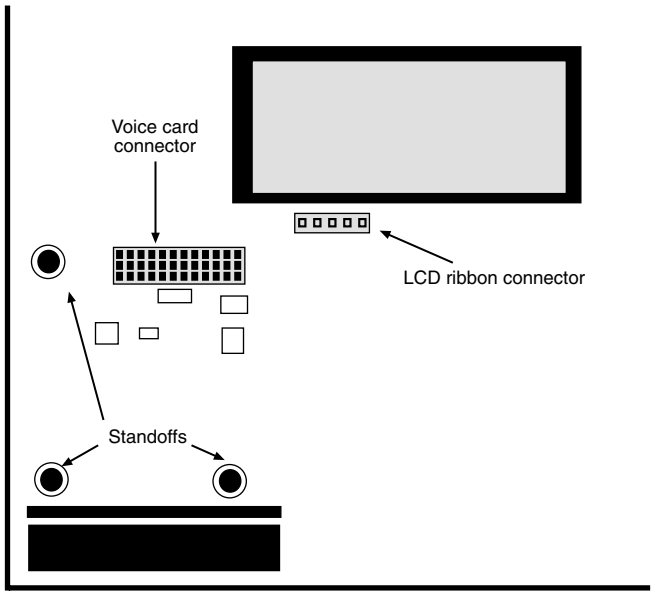


Figure 8: Voice board mounting on the main board

Line the voice board up with the connector and push it carefully but firmly into place. With the three screws provided with the voice card, secure it to the three standoffs.

4. Replace the cover, manually reconnecting the keypad ribbon to the board. Replace the four screws. Your voice board is now connected and ready for use.

See Chapter 11 for information on recording your own messages.

Chapter 4: POWER SUPPLY & BATTERY BACKUP

POWER REQUIREMENTS

The SCADA 3000 is designed to be powered from a 10-15V DC power supply (15V required to charge an external battery). This wide operating voltage range permits the unit to be powered from 12V solar cells in truly remote applications. Most applications will have 120VAC available and we recommend using one of the 15VDC power supplies listed below:

FGD-3100	SCADA 3000 60W (15V, 4A) Hard-wired power supply
FGD-3110	SCADA 3000 50W (15V, 3.3A) Plug-in power supply with cord

Both supplies will accept inputs from 90-264 VAC 47/63Hz. The FGD-3100 has terminals for hardwiring AC power. The FGD-3110 requires a power cord with an IEC connector on the end (included).

The SCADA 3000 Main unit includes an Aux Power output specifically for powering SCADA 3000 expansion modules. The Aux Power terminal is backed up by the external battery.

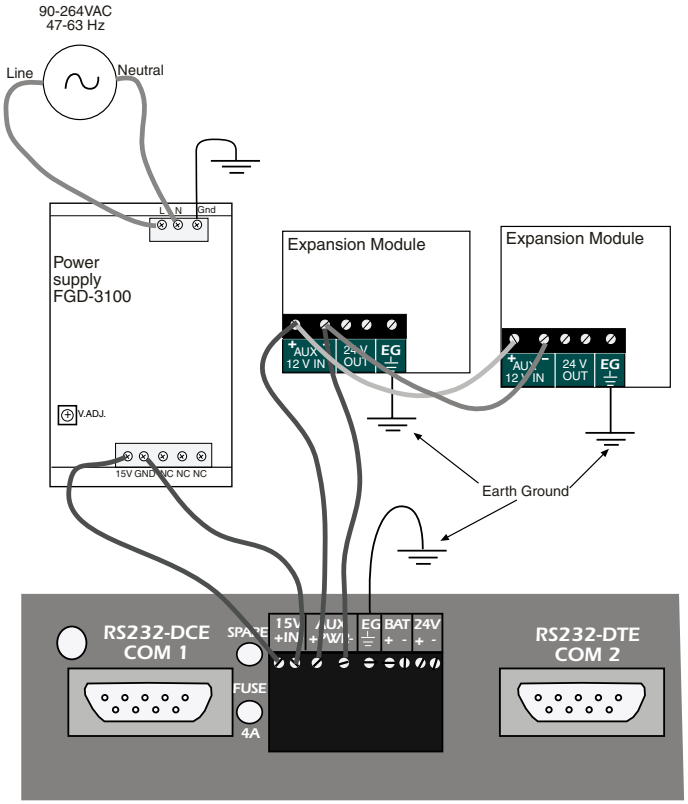


Figure 1: Wiring from Power Supply to SCADA 3000

GROUNDING AND POWER SURGE PROTECTION

SCADA 3000 should be earth grounded by connecting a true earth ground to the terminal labeled EG. This is essential to protect SCADA 3000 from possible damage due to static electricity, power surges and lightning strikes.

The Sensaphone® SCADA 3000 can be damaged by power surges and lightning through the telephone line and the power supply. Although SCADA 3000 has built-in surge protection, consideration should be given to provide additional protection for the unit and for any electronic equipment that is attached to your power supply and telephone lines, especially if you live in a lightning-prone area. Surge protection is available through Phonetics. Call for details.

POWER SUPPLY FUSE

The main unit includes a 4A Fuse to protect the unit and power supply from short circuits and/or product malfunction. If no lights come on when you depress the power switch, it is possible that the fuse is blown. You can verify a blown fuse by removing it and testing for continuity. To remove the fuse, simply pull it straight out of its socket. A spare fuse has been included for convenience, but be sure to correct the problem before trying another fuse (Replacement fuse: Wickman #3721400041). Any of the following could cause the fuse to blow:

- too much current being drawn due to the number of modules connected.
- voltage at the power input terminals too high (16V Max.)
- internal circuit malfunction

ESTIMATING POWER CONSUMPTION

Determining the power consumption of the system is required to make sure you have enough power available from your power supply in addition to determining the proper size battery for your application.

Listed below is the typical power consumed by the main unit and each module in the SCADA 3000 family when powered from 15 VDC.

Calculating Power Consumption for the SCADA 3000

	Quantity	x	Current (Ea.)	=	Total (amps)
Main RTU power	_____	x	0.215 Amps	=	_____
Modem option	_____	x	0.085 Amps	=	_____
Real Voice option	_____	x	0.008 Amps	=	_____
Universal input module	_____	x	0.100 Amps	=	_____
Pulse Count module	_____	x	0.100 Amps	=	_____
Thermocouple module	_____	x	0.100 Amps	=	_____
Relay output module	_____	x	0.080 Amps	=	_____
Analog output module	_____	x	0.080 Amps	=	_____
Annunciator module	_____	x	0.100 Amps	=	_____
Battery Charge Current	_____	x	0.600 Amps	=	_____
*Number of 4-20mA transducers	_____	x	0.038 Amps	=	_____

powered from the 24 V supply

Add Wireless Data Radio ____ Amps(avg.) _____

Total Current = _____ Amps

Compute the Total Power by multiplying the Total Current by the Supply Voltage

TOTAL POWER (watts) = 15V x Total Current = _____ (Watts)

•This Watt rating is the average power that the system will consume during normal operation; however, during startup the power is significantly greater.

*The 24V supply on the CPU and modules may be used to power 4-20mA transducers; be sure to account for these in your calculation by entering the number of transducers connected.

A sample computation follows:

Main RTU power*				00.215 Amps
Modem option	1	x	0.085 Amps =	00.085
Real Voice option	1	x	0.008 Amps =	00.008
Universal input module*	2	x	0.100 Amps =	00.200
Relay output module	1	x	0.080 Amps =	00.080
Battery Charge Current	1	x	0.600 Amps =	00.600
Powered 4-20mA transducers	3	x	0.038 Amps =	00.114
			Total Current	1.302 Amps

Warning: The Total Current must not exceed 3.5 Amps.*

Total Power (Watts) = 15 x 1.302 Amps = 19.53 Watts

*Exceeding this value will bring the system very close to the rated current of the fuse (4A). A small amount of margin is required to insure that the fuse doesn't blow under normal operation.

Power Supply Selection

To size the power supply properly, multiply the Total Power computed above by 1.5. This will insure that the power supply will have enough reserve power to operate properly during peak power demand.

Example: If the Total Power was 19.53 Watts, then the Power Supply should be rated at:

19.53 Watts x 1.5 = 29.30 Watts, or **30 Watts.**

Phonetics SCADA 3000 Power Supplies:

FGD-3100 Hard-wired power supply: 60 Watts

FGD-3110 Plug-in power supply: 50 Watts

(others are available)

BACKUP BATTERY (OPTIONAL)

In the event of a power failure, SCADA 3000 can continue to operate if connected to a sealed lead-acid gel-cell rechargeable battery. The capacity or Amp-Hour (AH) rating of the battery may vary between 5 and 18 AH.

To compute the appropriate size battery, multiply the backup time required by the Total Current.

Example: Battery Capacity = Hours of backup x Total Current

= 12 hours x 1.302 Amps = 15.62 AH

In this example, a battery with a capacity of at least 15.6 AH is required to provide 12 hours of backup time. Note that actual backup time will vary somewhat depending on temperature, battery age, and dialing activity. The battery connects to the terminals labeled BAT + and - on the SCADA 3000. (See Figure 2 below)

Note: Only connect the battery while the Power Switch is OFF.

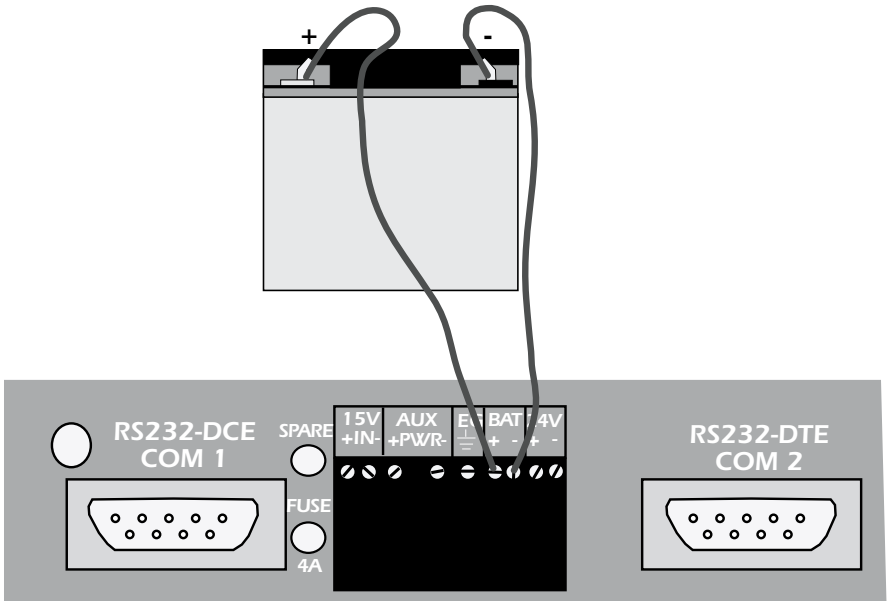


Figure 2: Wiring from optional battery to SCADA 3000

To determine battery backup time: $\frac{\text{Battery capacity(AH)}}{\text{Total Current (A)}} = \frac{18\text{ AH}}{1.302\text{ A}} =$ **13.8 hrs.**

Whenever the power switch is turned on and the power supply is plugged in, SCADA 3000 will automatically charge the battery. The unit also includes special circuity to prevent the battery from being damaged in the event of an extended power outage. When the battery runs down to 9V, the unit will automatically disconnect it, preventing deep-discharge damage. The battery will remain disconnected until main power returns. Note that, once the deep-discharge circuit has been activated, the unit requires power at the 15V terminals to turn back on, even if the 12V battery is fully charged. The battery should provide 5 years of service before needing replacement.

NOTE: Have battery serviced by qualified service personnel only.

POWERING THE SCADA 3000 WITH SOLAR CELLS

The SCADA 3000 is a relatively low-power device (2-4 Watts, depending on options) and is suitable for operation from solar cells. Typically solar power is only used when AC power is unavailable or too costly to have service installed at the unit's location. Proper component selection is critical to keep the unit functioning during extended periods of darkness or poor sunlight intensity. Critical applications should overcompensate on capacity to insure that the unit will continue to function. Component selection will vary depending on your geographic location. Regions with long periods of low intensity sunlight may not be suitable for solar operation. Consult with a solar power specialist for specific questions on component selection and geographic issues.

Providing reliable power from solar cells requires (3) components: one or more solar panels, one or more large batteries, and a solar battery charger/regulator. During periods of sunlight the solar cells will provide enough energy to power the SCADA 3000 and charge the batteries. When there is no sunlight the batteries will power the unit. The figure below illustrates the connection between the components:

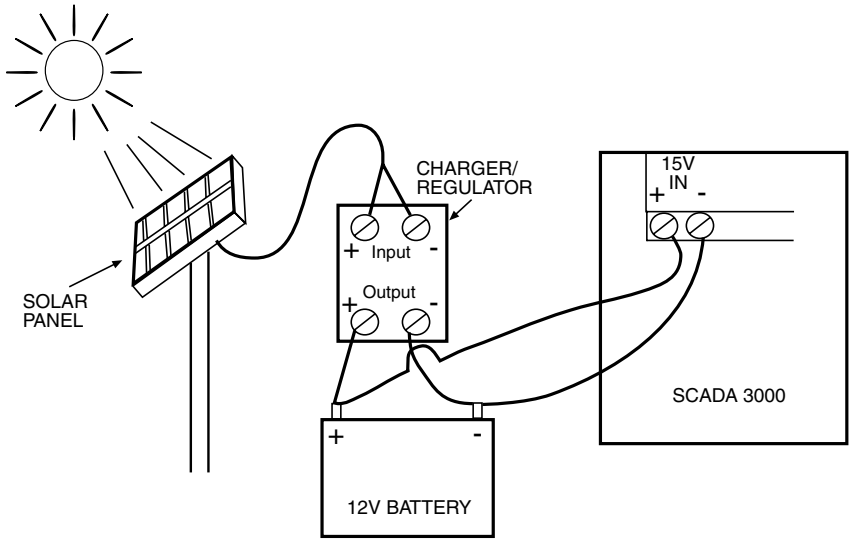


Figure 3: Solar Power wiring

The SCADA 3000 requires 10-15 Volts DC to operate. There are many 12V solar panels available which will meet this requirement. Current consumption will depend on installed options such as: voice/modem boards, I/O modules, radio/modem or cellular phone. The batteries are usually similar to automobile or marine type batteries. Many solar panel dealers also carry chargers and batteries and can assist in selecting a matching component system. It is important that the components be designed as a system, otherwise component damage or power loss may result.

To size the solar panel requirements, use the “Estimating Power Consumption” worksheet a few pages back, but adjust the value for the Battery Charge Current since the solar panel will charge the battery and not the SCADA 3000 charge circuit.

Chapter 5: INPUT WIRING & CONFIGURATION

The SCADA 3000 is compatible with a wide variety of electrical signal sources including contacts, 10K thermistors, analog voltage outputs and 4-20mA current sources. Each type of signal requires the configuration jumpers, located behind the input terminal strip, to be installed in the proper position for each type of signal. Follow the instructions below to properly wire and configure the inputs for each type of electrical signal. Note: Needlenose pliers will be required to move the jumpers.

> Warning: The inputs are designed to work with low voltage signals from 0-5V. **DO NOT** connect differential voltages greater than 5V to the inputs. **DO NOT** connect 120VAC to the inputs. In the 4-20mA jumper position and when no jumper is installed, the inputs are differential and will work with signals that are +/-12V away from power supply ground. This allows the use of multiple 4-20mA circuits on one loop. See circuit schematics for more information.

General Wiring Considerations

Most dry contact sensors can be connected to the SCADA 3000 using inexpensive 2-conductor twisted-pair cable as small as #24 AWG. For thermistor, 0-5V and 4-20mA sensors, use the wire chart below as a reference for selecting the appropriate wire gauge. Note that if the sensor is located far from the unit or if you are running cable in an electrically noisy environment, you should seriously consider using twisted pair shielded cable. This will shield the signal from electrical interference, thereby preventing false readings and/or damage to the unit. If you are using shielded cable, connect the shield to the EG pin on the unit. To minimize electrical noise coupling between I/O lines, follow the guidelines listed below for routing your wires:

- Route the power supply and communications cables to the unit by a separate path from wiring to I/O devices. Where paths must cross, their intersection should be perpendicular.
- Do not run signal or communications wiring and power in the same conduit.
- Segregate I/O wiring by signal type. Bundle wiring with similar electrical characteristics together.
- Allow at least two inches between the unit and I/O wiring ducts.

Wiring Distance	Minimum Wire Gauge
700'	#24 AWG
1500'	#22 AWG
2500'	#20 AWG

TERMINAL BLOCKS

The terminal blocks for wiring the inputs (and outputs) are removable. They should only be removed or attached when power to the SCADA 3000 is OFF. If for instance you had to return your SCADA 3000 for service, you could remove the terminal blocks in order to maintain your

current wiring configuration. To remove terminal blocks use your thumb or forefinger to push the screw section of the block straight out. (See Figure 1)

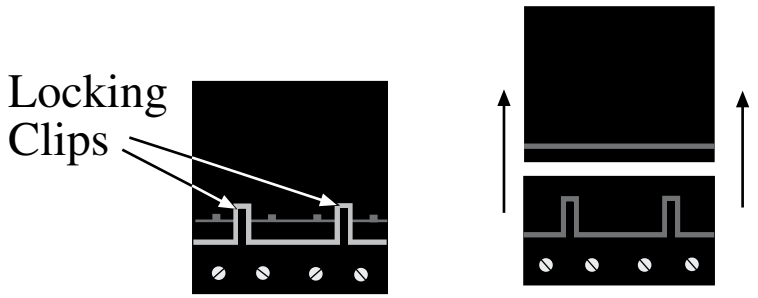


Figure 1: Removing terminal blocks

INPUT CONFIGURATION

Correct Jumper Positioning

It is extremely important that you position the jumpers correctly on the SCADA 3000. The input jumpers are located directly behind the Input terminal strip. There is one jumper for each input (16 total). Note that each jumper connects four pins. If you look closely at the jumpers, you will note that each one reveals two thin metal strips. When the jumper is fit into place, the strips contact the header pins corresponding to the input you are setting. **The strips must be positioned vertically**, not horizontally, for your inputs to work. (See illustration below.) Please be particularly careful when positioning the jumpers.

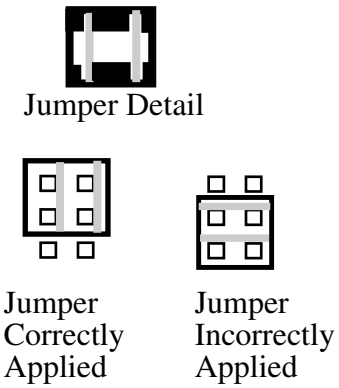


Figure 2: Correct Jumper positioning

Normally Open/Normally Closed Dry Contacts

Dry contact sources consist of relays or switches that are isolated and have no external voltage applied. These devices can be connected directly to the input terminals without regard for polarity. Note that Solid State outputs may be polarity sensitive. If using Solid State outputs, observe the polarity markings next to the terminal strip on the SCADA 3000. Choose an input and connect the wires to the corresponding screw terminals for that input. The configuration jumper should be set to the TMP/DRY position. The following figure shows how to connect a dry contact sensor:

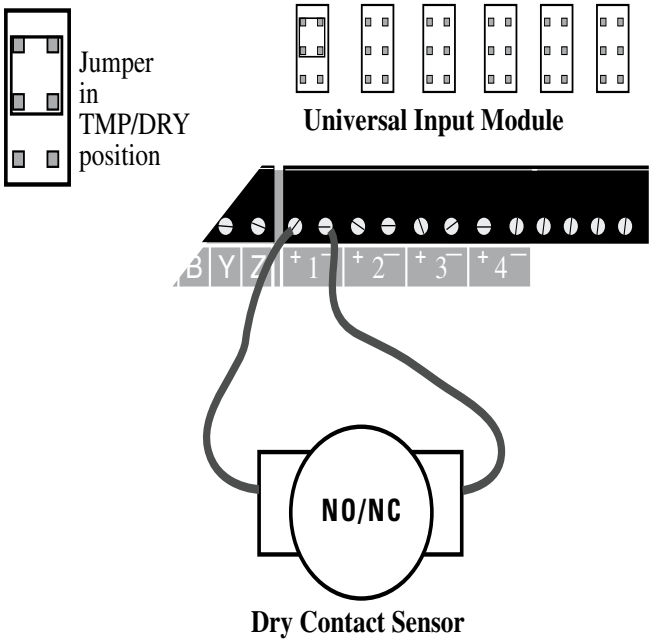


Figure 3: Wiring a Dry Contact Sensor

The shunt set to the TMP/DRY position is reflected in the following circuit diagram.

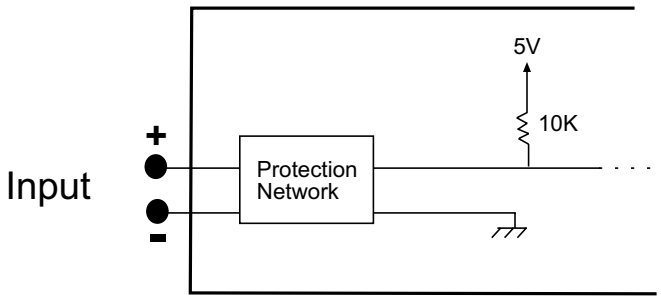


Figure 4: TMP/DRY Circuit

1OK Thermistors

The SCADA 3000 is compatible with 10K thermistors that match the curve data listed in the tables in Appendix C. The monitoring temperature range of the 10K thermistor is -80 to 300ºF (-62º to 149ºC). Thermistors can be connected directly to the input terminals without regard for polarity. Choose an input and connect the wires to the corresponding screw terminals for that input. The configuration jumper should be set to the TMP/DRY position. The figure below shows how to connect a thermistor sensor:

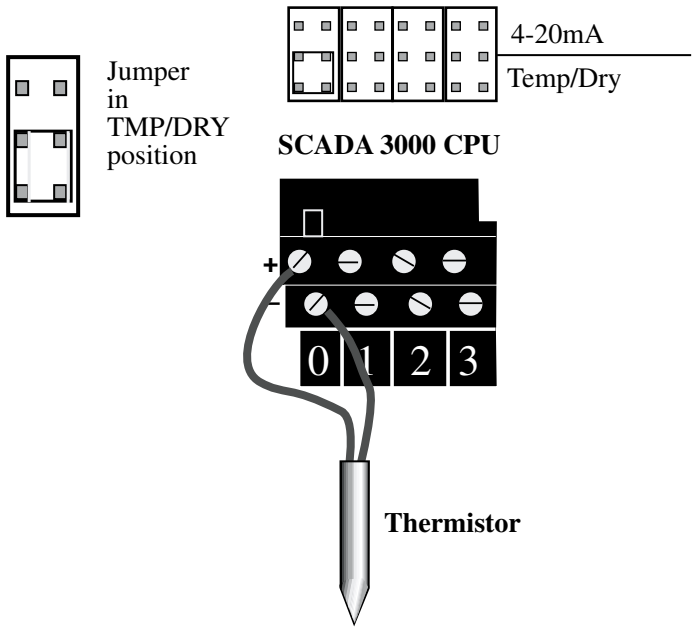


Figure 5: Wiring a Thermistor

Analog Voltage Output Transducers (0-5VDC)

The SCADA 3000 is compatible with transducers that produce an analog output of 0 to 5VDC. Analog voltage output devices can be connected directly to the input terminals but be sure to follow the polarity markings on the unit. Choose an input and connect the wires to the corresponding screw terminals for that input. The configuration jumper should be set to the storage position as indicated in the diagram below:

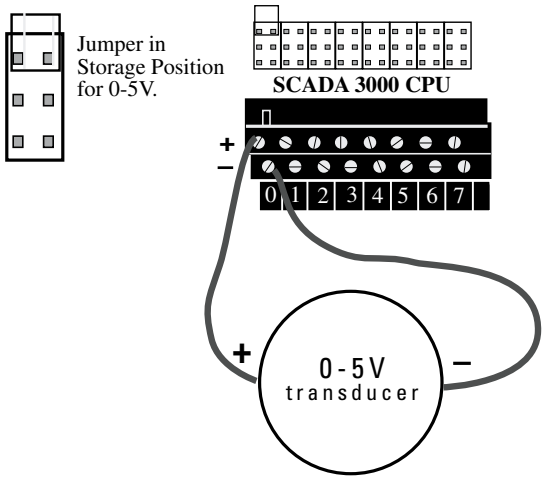


Figure 6: Wiring a 0-5V transducer

The shunt set to the 0-5V position is reflected in the following circuit diagram.

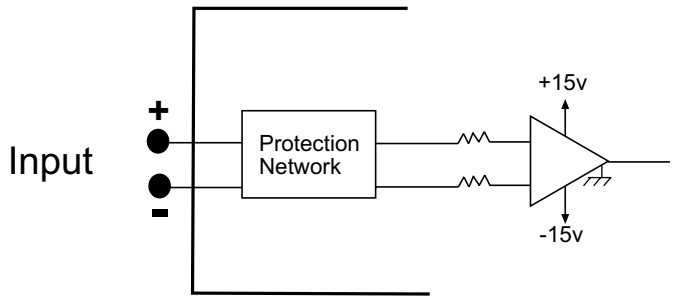


Figure 7: 0-5V Circuit diagram

4-20mA Current Loop Transducers

The inputs on the SCADA 3000 are compatible with transducers that produce an analog output current of 4 to 20mA. Analog 4-20mA output devices sometimes require a 24VDC power supply to operate. The SCADA 3000 provides a built-in 24VDC power supply for this purpose. The 24VDC power supply has enough capacity to power up to sixteen 4-20mA devices. Follow the wiring diagrams below for connecting a 4-20mA device. The configuration jumper should be set to the 4-20mA position as indicated in the following diagrams:

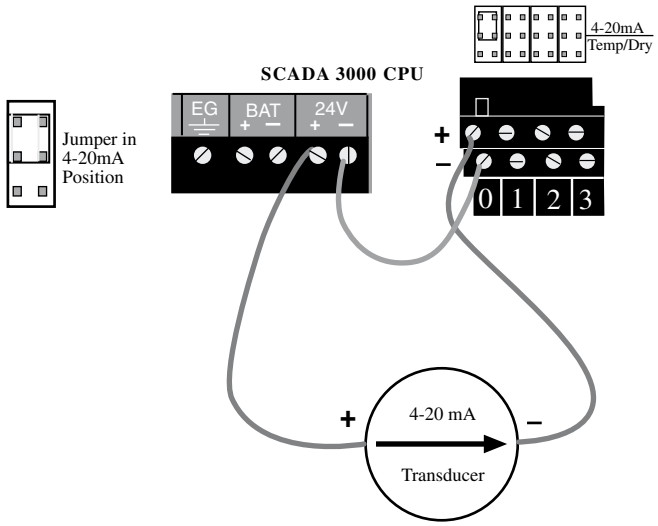


Figure 8: Wiring a 4-20mA device using the built-in 24 VDC supply.

The shunt set to the 4-20mA position is reflected in the following circuit diagram.

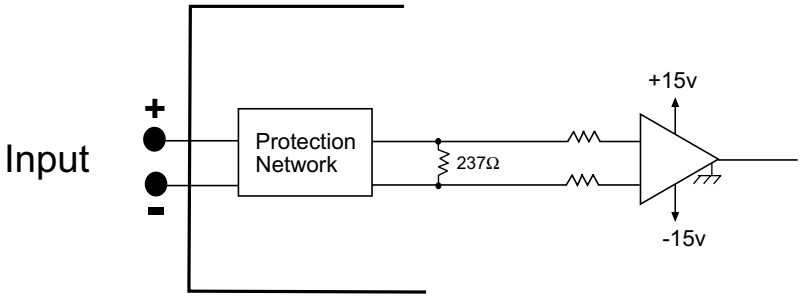


Figure 9: 4-20mA Circuit diagram

If, for any reason, you choose to utilize a separate power supply, your wiring configuration from the SCADA 3000 should look like the following diagram:

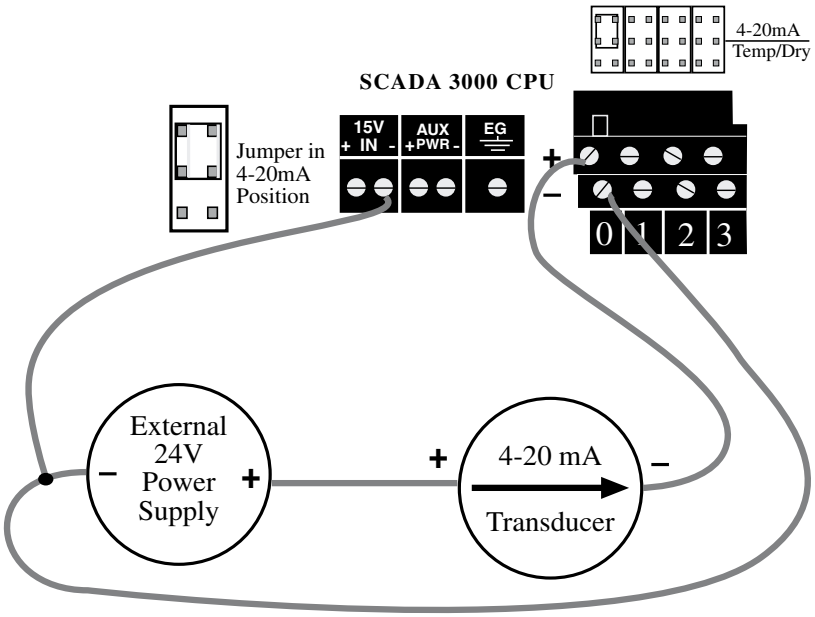


Figure 10: Wiring a 4-20mA device using an external 24 VDC supply.

Connect powered 4-20mA devices according to the diagram below.

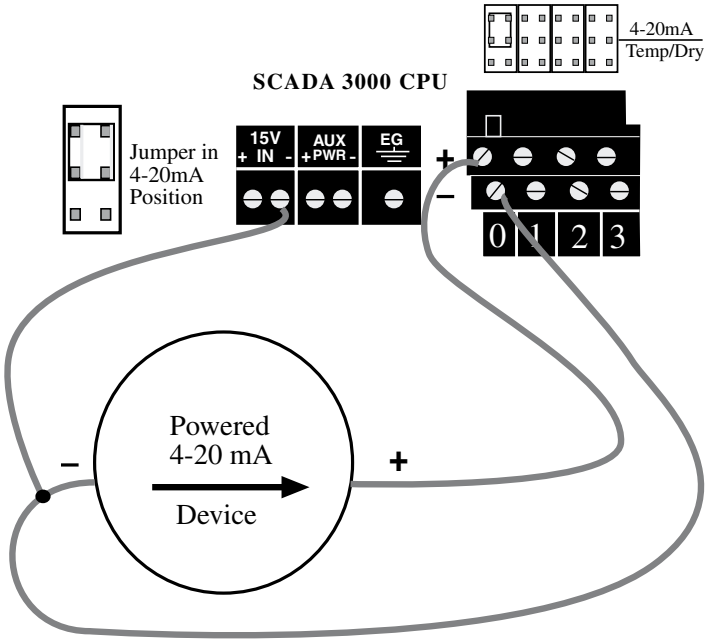


Figure 11: Wiring a 4-20mA Powered Device

Chapter 6: OUTPUT WIRING

The SCADA 3000 provides eight latching relay outputs that can be used to switch equipment on and off. The relays are capable of switching up to 2A at 125VAC and have integral contact protection. They may be controlled manually while on-line using the SCADA 3000 package, or automatically via a ladder program or C-program. LED indicators are provided to show the ON/OFF status of each relay output. Output relays are a latching type and will continue to hold their state without power.

NOTES ON CORRECT USE

The Outputs can be used to control a wide variety of AC and DC devices. However, care must be taken to insure that the SCADA 3000 is not used in such a way that will severely degrade the life expectancy of the outputs or cause the unit to malfunction (e.g. microprocessor reset). Do not continuously apply a voltage higher than the rated maximum voltage to any output. Do not switch loads that exceed the rated capacity of the output. Avoid switching high-current inductive loads as these will generate significant radiated noise when switching, possibly causing the microprocessor to reset. Also, do not run your output wiring together with your input wiring. This will almost always cause errors in your input readings because of signal coupling. Be sure to follow National Electric Code guidelines for proper wiring and safety when working with high voltage circuits.

The wiring diagram below shows a typical switching application:

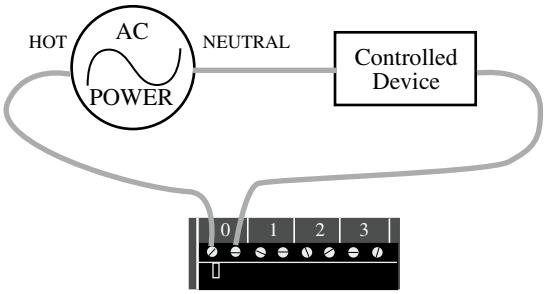


Figure 1: Output Wiring

The output wiring terminals are designed to accept #14AWG to #24AWG stranded wires. Select the appropriate cable based on the signal or load you will be switching. Follow all NEC codes and guidelines as well as local codes required to maintain safety when working with high voltages. To minimize electrical noise coupling between I/O lines, follow the guidelines listed below for routing your wires:

- Route output wiring to the unit by a separate path from the input and communication cables.
- Do not run output wiring in the same conduit with input and communications wiring.
- Segregate I/O wiring by signal type. Bundle wiring with similar electrical characteristics together.
- Allow at least two inches between the unit and I/O wiring ducts.

The terminal blocks for wiring the outputs (and inputs) are removable. They should only be removed or attached when power to the SCADA 3000 is OFF. If for instance you had to return your SCADA 3000 for service, you could remove the terminal blocks in order to maintain your current wiring configuration. To remove terminal blocks use your thumb or forefinger to push the screw section of the block straight out. (see figure below).

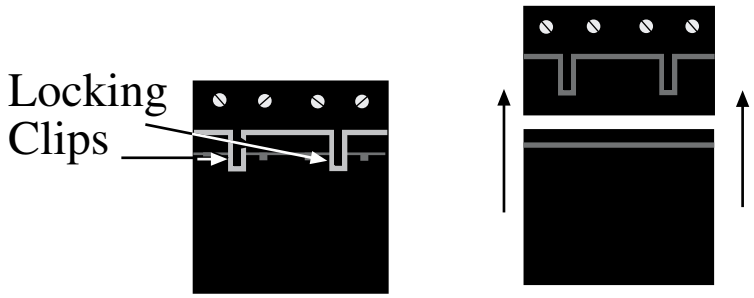


Figure 2: Removing terminal blocks

Chapter 7: I/O EXPANSION MODULES

ADDING EXPANSION MODULES

The capabilities of the SCADA 3000 can be expanded by connecting additional I/O modules to the main unit. This allows the system to monitor or control up to 144 points. A communications cable and a power cable must be routed to each module. Modules can be located up to 2000 feet from the main unit. Battery-backed power is provided for the modules; however, care must be taken not to exceed the maximum power limit of the power supply.

When adding an I/O module, be sure to turn OFF power to the SCADA 3000 during all wiring and installation. When powered up, the unit will automatically check to see if any modules are connected.

Cabling Notes

Follow these guidelines to protect the communications cable from electrical interference.

- Keep the communications cable at least five feet from any electric motors, transformers, rectifiers, generators, arc welders, induction furnaces, or sources of microwave radiation.
- If you must run the cable across power lines, run them at right angles to the lines.
- Keep the communications cable at least 6 inches from AC power lines carrying less than 20A, at least 1 foot from lines greater than 20A, and 2 feet from lines greater than 100KVA.
- If you run the cable in a metallic wireway or conduit, keep the communications cable at least 3 inches from AC power lines carrying less than 20A, at least 6 inches from lines greater than 20A, and 1 foot from lines greater than 100KVA.

Running the communications cable through conduit provides extra protection from physical damage and electrical interference. If you route the cable through conduit be sure to ground the entire length by connecting it to the building earth ground. For best results, use ferromagnetic conduit near critical sources of electrical interference. Aluminum conduit may be used in non-critical areas.

MODULES

Up to 15 expansion modules may be connected to the SCADA 3000 main unit to provide additional inputs and/or outputs. Each module must be configured with its own *unique* address using the BUS ADDRESS jumpers (see module documentation). You may mix & match modules to suit your application's requirements. Specific modules are available to connect to various types of sensors and equipment. The modules offered by Phonetics are listed as follows:

FGD-3010	8-Channel Universal Analog Input Module
FGD-3020	4-Channel Pulse Count Module
FGD-3030	4-Channel Thermocouple Module
FGD-3040	8-Channel Relay Output Module
FGD-3050	4-Channel Analog Output Module (4-20mA, 0-20mA, 0-10V)
FGD-3060	Annunciator Module

Power Supply

The modules require 10-15VDC to operate. A 14V power supply (labeled AUX PWR) is provided on the main unit for powering the modules. The AUX PWR terminals are battery-backed from the main battery connected to the SCADA 3000. Be sure you don't exceed the maximum current available from your main power supply. The main board is fused at 4 Amps. Alternately, you may power the modules from any 10-15VDC power source, but to prevent data transmission errors be sure to connect the earth ground terminal to a good electrical earth ground or cold water pipe. (See Figure 1)

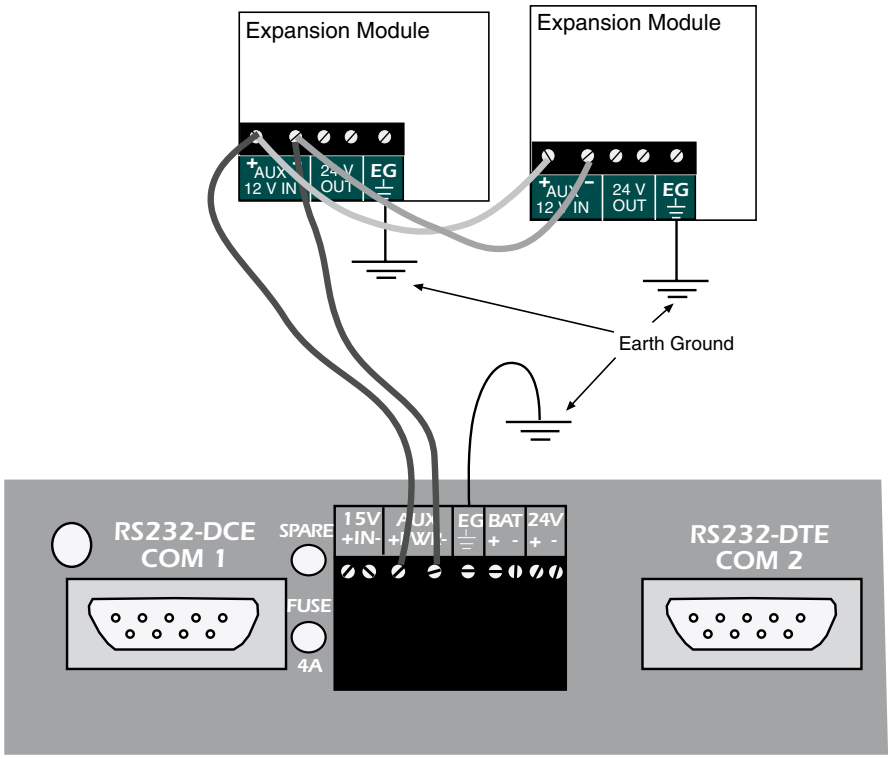


Figure 1: Main Unit Power to Modules

COMMUNICATIONS BUS WIRING

The modules communicate with the main unit via a 4-wire cable connected to the Bus terminal block located on the bottom right side of the unit. (See Figure 2)

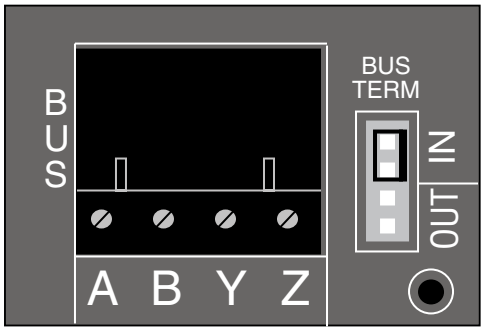


Figure 2: Bus terminal block

The cable used to connect the modules must be 4-conductor twisted pair with a nominal impedance of 120Ω (for example, Phonetics part number FGD-0051, Belden 8132 or 9842 cable) and should be routed away from AC cables and power supplies. Use one pair for A & B and the other pair for Z & Y. Figure 3 below shows how to connect the cable between the main unit and the modules. Note how the wiring is reversed between the modules and the main unit, but between modules the wiring is straight through.

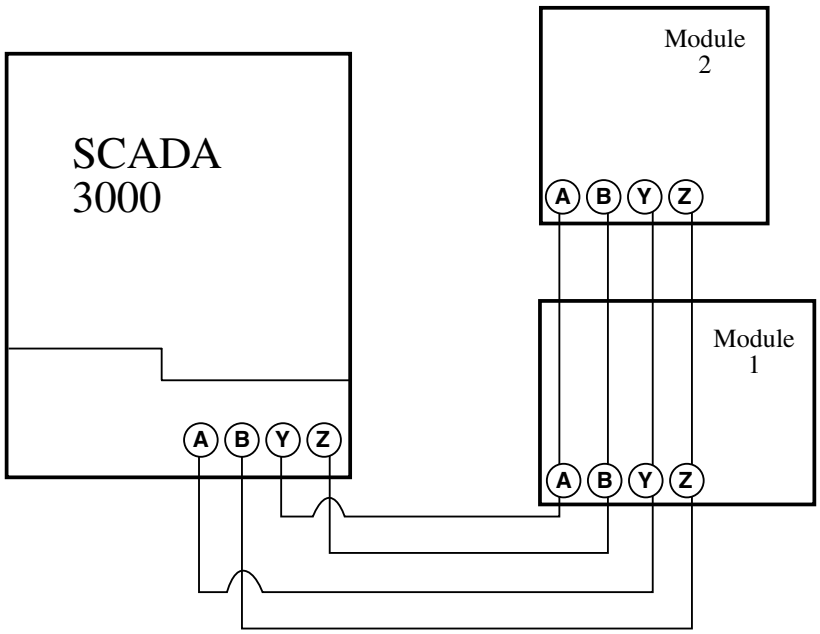


Figure 3: Communications Bus Wiring

The modules may be located up to 2000 feet away from the main unit. Do not run AC power lines in the same cable or conduit as the communications wiring, as this can cause data errors.

Bus Termination

Both the SCADA 3000 main unit and the final module in the chain must be terminated. A termination jumper is located on each device. The modules (if any) branching off the main connection are not terminated. While the main connection between the SCADA 3000 and the most distant module can be 2000 ft., the individual modules in-between are wired on branches called Stub lines. Stub lines can be no more than 50 ft. from the main line. (See Figure following)

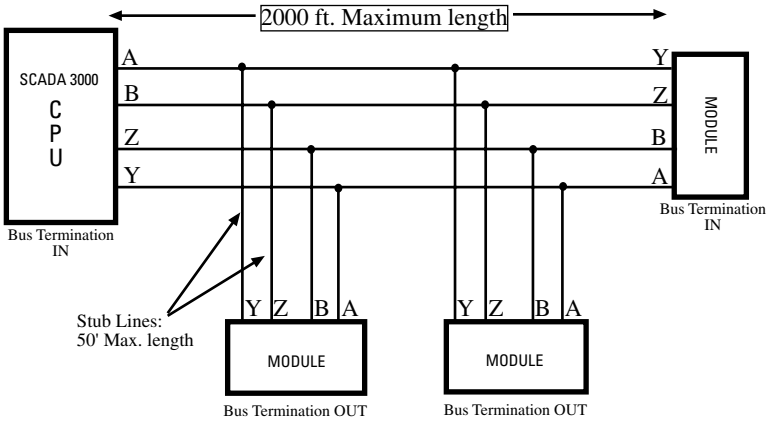


Figure 4: Communication Cable Wiring and Bus Termination

Module Types

Both input and output modules can be connected to the SCADA 3000. The SCADA 3000 can operate with as many as 144 I/O points--that is, up to 144 functioning inputs and/or outputs. Phonetics offers three different types of input and three types of output modules.

Input Modules

UNIVERSAL INPUT MODULE: This module provides 8 Universal Inputs. They are identical to the 16 inputs on the SCADA 3000 main unit. Inputs can be set up and wired as NO/NC dry contact, thermistor, analog voltage input, or 4-20mA transducer.

PULSE COUNT MODULE: This module will count contact closures from a dry circuit at a rate of up to 10,800 pulses per second. There are four pulse count channels per module. In addition, the pulse rate for each channel will be computed on a per second, per minute, or per hour basis. If a rate computation is not desired, you may also select totalizer overflow to accumulate pulses that exceed 1 million. The most common uses for a pulse count module include flow rates and rpms.

THERMOCOUPLE MODULE: The Thermocouple module offers 4 inputs for thermocouple temperature sensors. Thermocouples differ from thermistors in that they can measure a much wider range of temperatures, such as those encountered in cryogenic freezer and industrial oven temperature measurement applications. The SCADA 3000 Thermocouple expansion module features very accurate 16-Bit input resolution and is compatible with E, J, K, T, R and S type thermocouples.

Output Modules

RELAY OUTPUT: This unit offers 8 more output relays, identical to the eight on the main unit.

ANALOG OUTPUT: This module offers 4 analog output channels. The outputs have 12 bits of resolution and can be configured as 4-20mA, 0-20mA, or 0-10V. Most analog outputs are used for PID applications (looped information used to maintain conditions) as outlined in Chapter 17.

ANUNCIATOR: The annunciator module provides eight LED indicators in a 1/8 DIN panel-mount enclosure. The eight LED indicators may be programmed to represent the on/off status of any real or virtual parameter in the SCADA 3000.

Chapter 8: SOFTWARE INSTALLATION & SETUP

This section describes how to install and configure the SCADA 3000 Software for your computer and modem.

MINIMUM REQUIREMENTS

Intel Pentium processor or equivalent

20 MB of free disk space

32 MB of RAM (64 MB RAM recommended)

Graphics card and monitor compatible with Microsoft Windows® 95, 98 or NT (such as VGA)

Microsoft Windows® 95, 98 or NT

Mouse

Windows-compatible phone modem and/or RS232 ports

INSTALLING THE SOFTWARE

The SCADA 3000 Windows Software setup program makes installation quick and easy. Make sure that all Windows applications are closed before attempting to run setup. If you encounter problems during installation, please call Phonetics Technical Support at (610)558-2700. The SCADA 3000 Software for Windows will install to a directory named C:\Program Files\SCADA3000, unless you choose to change the directory name. The installation program will give you three installation choices: Typical, Compact, or Custom. For new installations choose Typical.

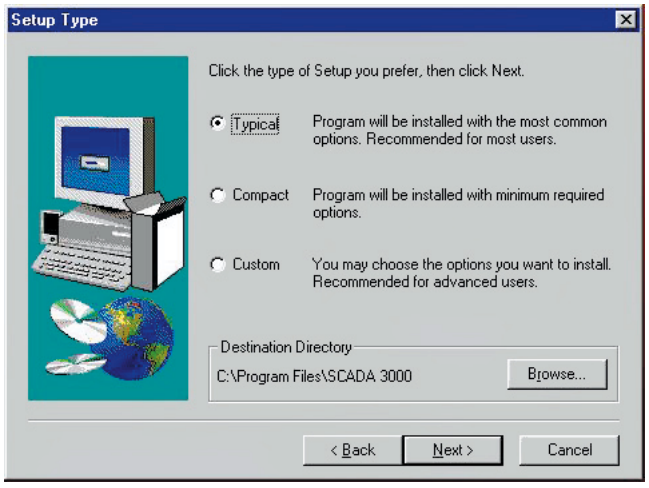

Setup Type
Click the type of Setup you prefer, then click Next.
Typical
Program will be installed with the most common options. Recommended for most users.
Compact
Program will be installed with minimum required options.
Custom
You may choose the options you want to install. Recommended for advanced users.
Destination Directory
C:\Program Files\SCADA 3000
Browse...
< Back
Next >
Cancel

Figure 1: Installation Choices form

PROCEDURE USING CD-ROM

1. Start Windows.
2. Insert SCADA 3000 CD-ROM
3. Select Start>Run and type in D:\Setup.exe and click OK.
   Note: If your CD-ROM is installed on a drive letter other than D, substitute the appropriate letter.
4. Follow the prompts until the installation is finished.

PROCEDURE USING DISKS

1. Start windows.
2. Insert SCADA 3000 Disk #1
3. Select Start>Run and type in A:\Setup.exe and click OK.
4. Follow the prompts, inserting each disk as prompted, until the installation is finished.
5. Insert the disk labeled SCADA 3000 Database Disk #1
6. Select Start>Run and type in A:\Setup.exe and click OK.
7. Follow the prompts, inserting each disk as prompted, until the installation is finished.

UPDATES VIA INTERNET

You can update your SCADA 3000 software and firmware via the Net-Upgrader program that comes installed with your SCADA 3000. Net-Upgrader allows future upgrades to be performed quickly and easily. Instructions for using the upgrader are included in the online Help files. To find out which software revision is currently installed on your computer, run the SCADA 3000 software and select Help>About.

For detailed information on the procedure for installing new firmware, see Appendix F in your owner's manual.

Chapter 9: COMMUNICATING WITH THE SCADA 3000

Running the Software

To run the software, double click the SCADA 3000 icon on your desktop. The Control Center form will appear. (See Figure 1)

The Control Center form is the main menu to access all of the features of the SCADA 3000.

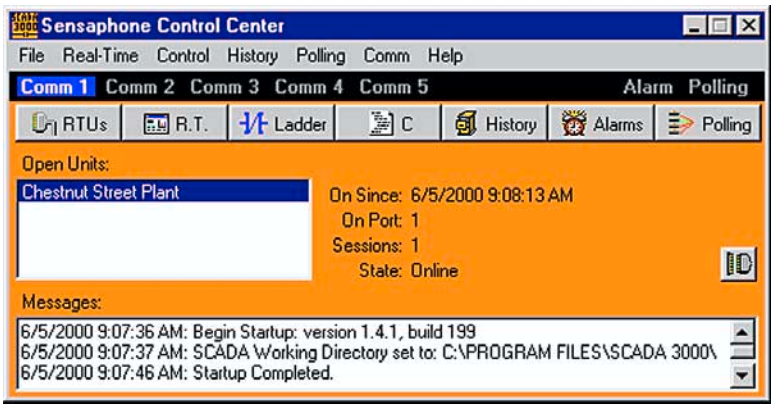

Sensaphone Control Center
File Real-Time Control History Polling Comm Help
Comm 1 Comm 2 Comm 3 Comm 4 Comm 5 Alarm Polling
RTUs R.T. Ladder C History Alarms Polling
Open Units:
Chestnut Street Plant
On Since: 6/5/2000 9:08:13 AM
On Port: 1
Sessions: 1
State: Online
Messages:
6/5/2000 9:07:36 AM: Begin Startup: version 1.4.1, build 199
6/5/2000 9:07:37 AM: SCADA Working Directory set to: C:\PROGRAM FILES\SCADA 3000\
6/5/2000 9:07:46 AM: Startup Completed.

Figure 1: SCADA 3000 Control Center screen

Communications Programming

The software will communicate to SCADA 3000 through your computer's local serial port, phone modem, or radio modem. These communication paths must be properly configured for a connection to be established.

To set up and configure ports manually, choose the COMM pulldown menu in the Control Center form, and then choose Setup. The following form will appear.

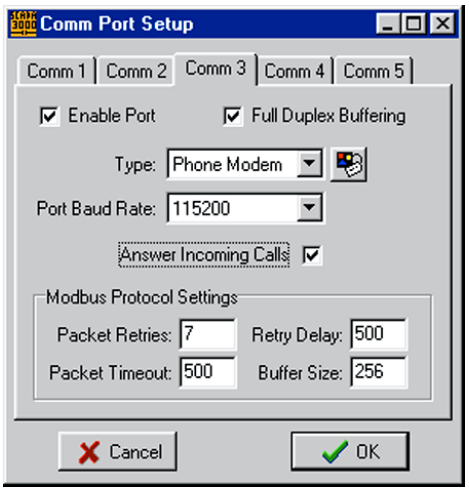


Figure 2: Comm Port Setup form

You can configure all four communication ports in one session if you prefer. For each port you're enabling, click in the "Enable Port" box. A description of the COMM port settings is provided below.

Type: Select the type of communication you will be using to communicate with the SCADA 3000, either serial port, phone modem, or radio modem.

Modem Setup Button: Click on this button to bring up the modem setup form shown below.

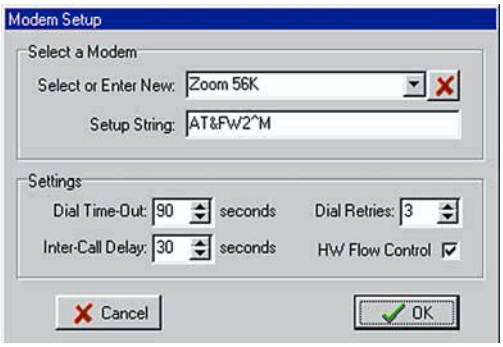


Figure 3: Modem Setup form

On this form, select the modem type and the necessary setup string for that modem. Change Dial Time-Out, the Inter-Call Delay, and Dial Retries if you want settings different from the defaults.

Baud Rate: Set the baud rate for the selected port. If your communication type is modem, set the baud rate to the fastest rate your modem will support. If you will be connecting your serial port directly to the SCADA 3000, set the baud rate to 38400.

Answer Incoming Calls: Check this box if you want the modem to receive alarm calls or e-mail calls from the SCADA 3000.

Full Duplex Buffering: Allows communications to run faster during Register Verification by buffering multiple bytes during transmission. Highly recommended for modem communications.

The Modbus Protocol Settings are explained below.

Packet Retries: Information flows back and forth between SCADA 3000 and your PC in data "packets." Both automatically check for packet transmission errors. When a bad packet is detected, it gets sent again. Packet Retries determines how many times a packet is sent before a communication error is logged. Default retries is 5.

Packet Timeout: The amount of time before the PC gives up and determines that an individual modbus request has timed out. Decreasing this parameter will not increase performance, but increasing the parameter may remove occasional errors. Default: 500 milliseconds.

Retry Delay: This is the amount of time the SCADA 3000 software will wait for a response from a unit before it decides to make another communication request. Default is 500 milliseconds.

Buffer Size: Determines the number of bytes to be buffered during communications. (typical=256)

When you finish configuring the ports manually, click *OK* to close this form.

Cellular Phone Communications

The SCADA 3000 may be connected to a cellular phone for applications where a telephone line is unavailable. Contact Phonetics for specific details. To improve modem communications over a cellular link, it is highly recommended that the host computer have a modem which supports enhanced cellular protocols commonly referred to as MNP10EC. Several modem manufacturers sell modems with this feature. However, you must verify that your modem supports MNP10EC to be absolutely sure. The SCADA 3000 modem will connect using this protocol if MNP10EC is enabled.

To enable the protocol in your modem, you must enter the Setup String shown below on the modem setup form. To access this form, select *Comm*, and then *Setup* from the Sensaphone Control Center Form. Next, select the Comm port corresponding to your modem. Click the Modem Setup button next to the Type field. The form below will appear.

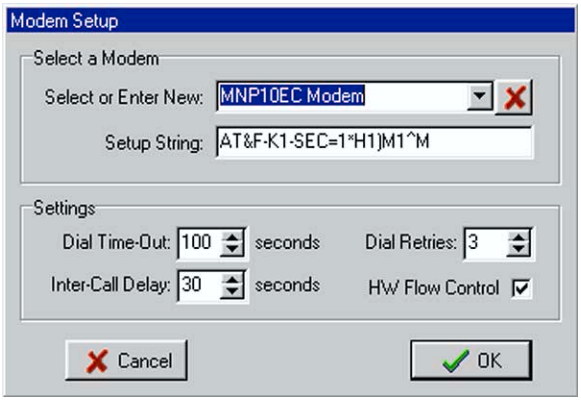


Figure 4: Modem Setup form

Enter the Setup String as shown above to activate the MNP10EC protocol in your modem, then click *OK*. Your modem should connect to the SCADA 3000 using this protocol when you have finished.

> **Note**: This setup string may have to change depending upon your modem manufacturer.

Creating New Units

To create a new unit, click on the RTUs button, or select *File>RTUs* from the menu on the Control Center screen. The following form will appear.

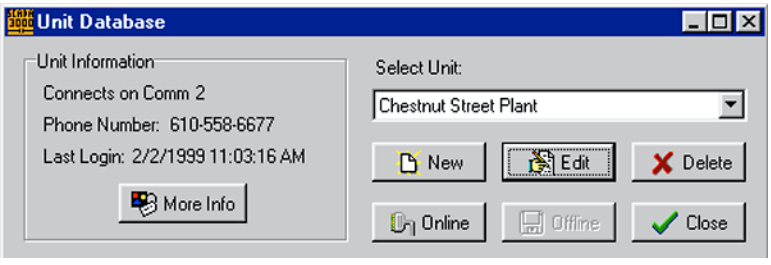


Figure 5: Unit DB form

Click on *New* to bring up the Create New Unit form. Type in a description for the individual unit, the phone number, and the slave ID. If you are on a Phone communication network and have only one unit per Comm port, the slave IDs can all be "1" with no conflict. However, if

you are using a Radio Communication network, each unit *must* have a unique slave ID. Select the Comm Port to which the unit is connected, then click OK. If you have more than one modem connected to your computer, then you may select *Any Phone Modem* and the software will select whichever modem is available.

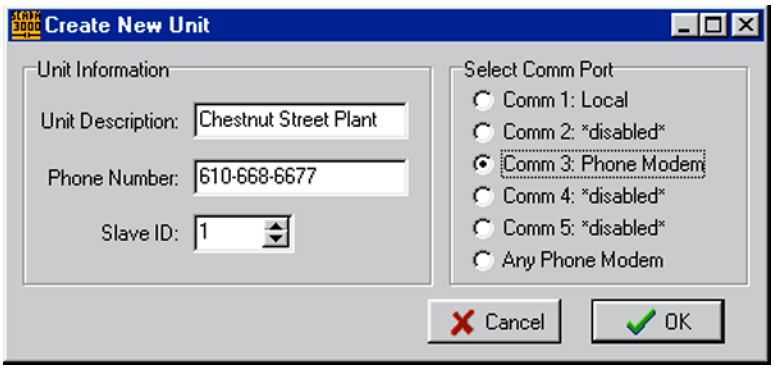


Figure 6: New Unit form

Communicating with the Unit

Return to the Unit Database form and click the Online button.

An interim Connecting form appears briefly as your computer connects to the unit.

The Main Programming form appears. You may now begin programming.

The Register Verification process will run in the background until it finishes comparing the software in the computer with that in the RTU. This procedure is required to ensure data integrity among multiple users of the unit. The Register Verification status bar shows the download progress and can be hidden or displayed at the user's discretion by clicking on *View>Register Verification* from the main menu and selecting *Show Register Verification*. If you choose not to view Register Verification, the clock showing Local Unit Time indicates if you are communicating with the RTU. If the time on the clock is not moving (advancing), then you are not in fact connected. If you view Register Verification at that point, you will not see any progress in the status bar.

System Security

The SCADA 3000 includes a programmable security option which provides 4 levels of password-secured user access to the system. Up to 64 multi-level user accounts and passwords can be set-up to allow personnel the appropriate level of system access. The 4 levels of user accounts are described below:

Account Types	Description
Administrator	A full-access account recommended for the person responsible for managing programming, & securing the system. This account provides full programming access, user-account/password management, and the ability to enable or disable the security option.
Programming	A full access account that allows complete programming capability, however, no user-account/password management is allowed.
Operator	A limited access account which permits the user to read/view any parameter, control outputs, enable/disable control programs, and acknowledge alarms. This account prohibits the ability to change programming or load programming.
Status-only	A read-only account which will allow users the ability to read/view any param-

eter or download the Datalogger, Event logger, C-program or Ladder program. This account will allow the user to change their password.

Disabled — An account set to disabled will prohibit the user from logging in. This can be used to temporarily restrict access without deleting the user account.

The system has the security option disabled as a default, which essentially gives any user full access to all system programming. Only a user with an Administrator account can enable security. Also, additional users can only be set-up by a user with an Administrator account.

To set-up an Administrator account and enable security, follow the steps below:

1) Go online with the SCADA 3000 and click on the *System* button from the toolbar or select *Programming* then *System* from the main menu. The following form will appear:

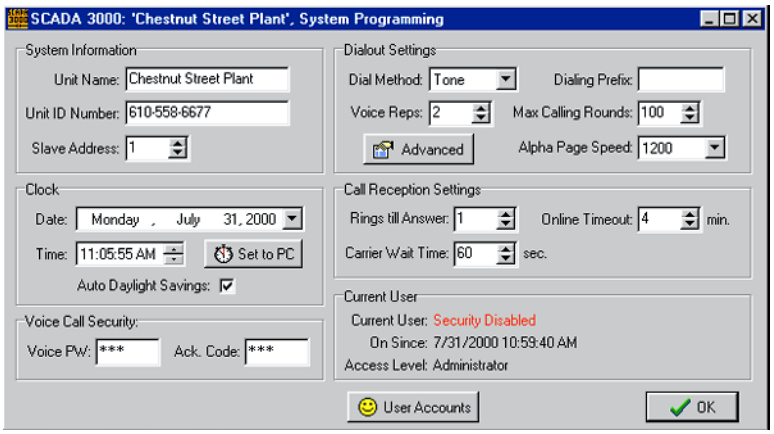


Figure 7: System Programming form

2) Click on *User Accounts*. The form below will appear:

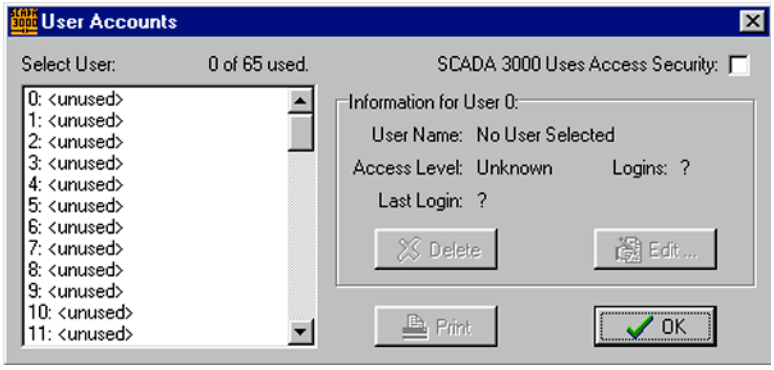


Figure 8: User Accounts form

Set up a new user account by clicking on the first *unused* account number and clicking the *Edit* button. The individual user account form will appear.

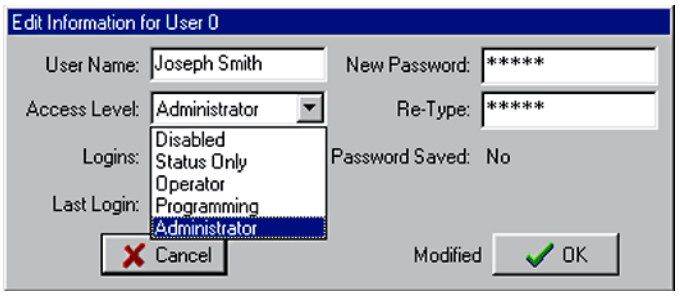


Figure 9: Edit Account Information form

This form will set up a user account for the SCADA 3000. Type in your *User Name*, select your *Access Level* by clicking the down arrow and choosing Administrator, and then enter a password in the *New Password* and *Re-Type* boxes. Click *OK*. An administrator account has now been created. To enable security, click on the *SCADA 3000 Uses Access Security* box on the User Accounts form. The next time you log in, you will need to know your user name and password.

User Log-In with Security Enabled

Once the user accounts have been configured and the security option enabled, users who attempt to log into the unit will be required to enter their user name and password every time. As an additional option, users may choose to save their password so that they need only click the OK button to log in. This option can be enabled on the unit log-in form shown below. To disable the remember-password option, uncheck the *Remember My Password* box on the *System* form.

Figure 10: Unit Log-on form

Setting Up User Accounts

Additional user accounts can only be set up by an Administrator. To set up a user account, first log in, then click on the *System* button from the toolbar or select *Programming*, followed by *System* from the main menu. Click on the *User Accounts* button. Select the next available user account and click the *Edit* button. Type in the *User Name*, select the *Access Level* by clicking the down arrow and choosing one of the options from the list. Then enter a password in the *New Password* and the *Re-Type* boxes. Click *OK*.

Deleting User Accounts

User accounts can only be deleted by an Administrator. To delete user accounts, log into the SCADA 3000 and click on the *System* button from the toolbar or select *Programming* and then *System* from the main menu. Click on the *User Accounts* button. Select the user account to be deleted and click on the *Delete* button. The software will ask if you are sure you want to delete the account. Click Yes to delete the account. Click *OK*.

Changing Passwords

A user's password may be changed with any level of security access once he or she has been logged into the unit. To change your password, click on the *System* button from the toolbar on

the main unit programming form, or select *Programming*>*System* from the main menu. Click on the *Change Password* button. The password change form will appear as shown below. Note that a user with an Administrator account may change the password for any user by selecting *User Accounts* from the system form and editing the user account information. This can be useful when personnel forget their password.

Figure 11: Change Password form

Radio Vs. Phone Communication

The SCADA 3000 software offers you the most flexibility when it comes to communicating with your SCADA 3000 RTUs. You can set up communication by data radio or phone line or a combination of both. The software allows you to configure up to four serial ports, which can be used for data radio and/or multiple phone modems.

Traditional SCADA devices generally communicate by radio. Radio allows the PC to be in constant communication with all connected RTUs all of the time. Real-time information is available on a continuous basis. In this mode the SCADA 3000 software will perform alarming and data logging from the PC.

When SCADA 3000 RTUs are connected via phone line the software can dial up each unit to retrieve and display information on an as-needed basis. In this mode, alarming and data logging are performed from the RTU itself.

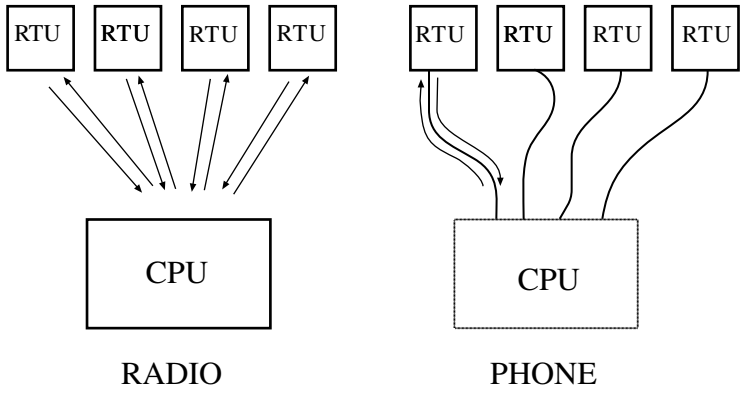


Figure 12: Radio vs. Phone communication

RS232 Communication Ports 1 & 2

The SCADA 3000 has two RS232 communication ports, one configured as DCE 9-pin female (COM 1) and the other as DTE 9-pin male (COM 2). COM 2 is used to communicate with SCADA 3000 via a radio modem (optional). The radio modem would connect to COM 2

directly with the appropriate cable. COM 1 is designed to connect directly to a personal computer for local communications/programming. The baud rate and communication type (radio or standard) for each port is selectable via jumpers located under the top cover (see Appendix D). The default Baud rates and communication types are listed below:

COM 1 Default setting: 38400 bps, Standard

COM 2 Default setting: 38400 bps, Standard

User Inactivity Timer

The User Inactivity Timer is an optional feature which is used to automatically disconnect a SCADA 3000 from an online session when the user has not actively moved the mouse or pressed any keys on the keyboard. The purpose is to prevent an unattended online session from occurring, because the SCADA 3000 will not dial out for alarms while an online session is in progress. To enable this feature, click on *File*, and then *Options* from the Sensaphone Control Center. Click the *Use Inactivity Timeout* box, set the desired timeout in minutes, and the click *OK*.

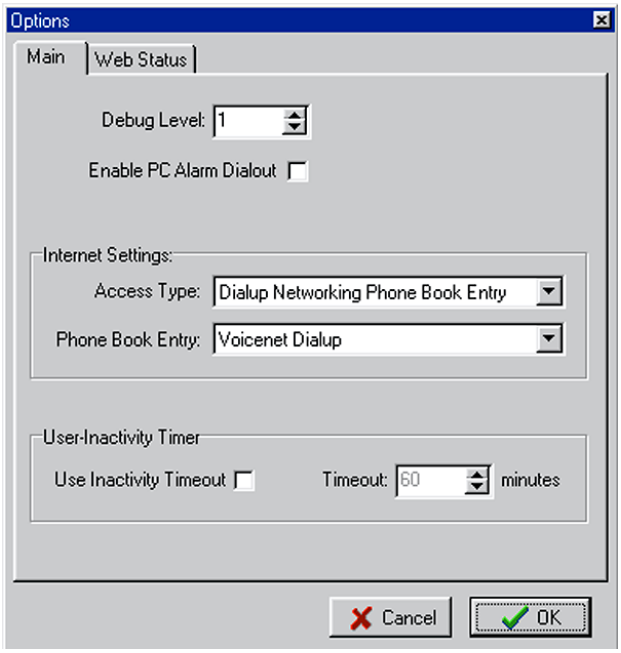


Figure 13: User Inactivity Timer

Chapter 10: GENERIC PROGRAMMING

This chapter includes information on how to program the following SCADA 3000 parameters that are constant regardless of communication type:

System
UAFs
I/O Points

SAVING AND LOADING PROGRAMMING FILES

All of the SCADA 3000 programming information, including expansion module programming, can be saved to one or more files. C programs and Ladder programs must be saved separately (See Chapters 17 and 18 for more information). By saving your programming, you will be able to reprogram the unit quickly in the event that the unit gets damaged and needs to be replaced. It is also convenient if you have multiple units that need to be programmed similarly, because you can load the same programming file into multiple units. To save your programming, click *File* and then *Save Programming* from the main menu while on-line with a unit. Enter a name for the file name and click *Save*. To load programming into a unit, go on-line and from the main menu click *File*, then *Load Programming*. Remember that C programs and Ladder programs must be loaded separately. Be sure to allow register verification to complete before loading or saving a programming file.

Note: Voice messages cannot be saved. These must be recorded into each individual unit.

CONNECTING

To access programming information in the SCADA 3000, double-click the SCADA 3000 software icon to bring up the Sensaphone Control Center form. Click on the RTUs button to bring up the Unit Database form. Select which unit you wish to program or click on the NEW button to create a new unit (see Chapter 9 for information on setting up a new unit). Click on the ONLINE button. The Connecting form will appear, followed by the Main Programming form (see figures 1 and 2a below).

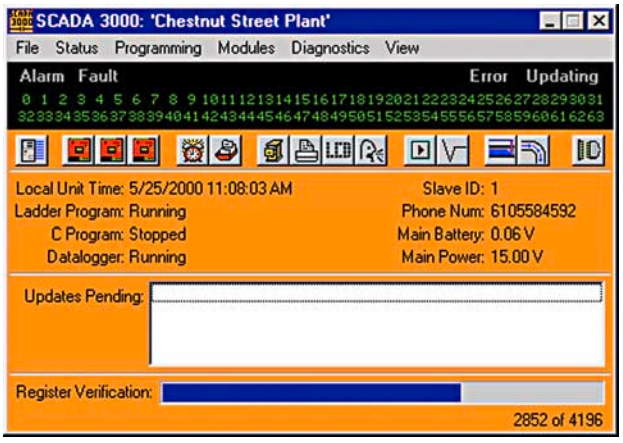


Figure 1a: Main Programming form, showing Register Verification

As soon as a connection is established, the software will begin Register Verification. This procedure is required to insure data integrity among multiple users of the unit. During this

process the software retrieves the programming from the unit. The Register Verification status bar shows the download progress and can be hidden or displayed at the user's discretion by clicking on *View* from the main menu and selecting *Show Register Verification*.

> **Note**: *Register Verification* will complete faster if you close the *Unit Status* window. To close the *Unit Status* window, select *View* from the main menu, then click on *Unit Status*.

The numbers on the I/O Module Programming buttons refer to the Module addresses 1-15. When an I/O expansion module is detected, the lower buttons will change to indicate that a module has been detected at the corresponding address. Two letters will appear on each button to identify the type of module as follows:

UI: Universal Input
PC: Pulse Count
TC: Thermocouple
RO: Relay Output
AO: Analog Output
AN: Annunciator

All expansion modules are detected automatically. If you have a module connected and it does not show up on this form, check the wiring to the module. Once the Main Programming form appears, you may begin selecting and programming from it. Register Verification will run automatically in the background while you begin programming; it compares the unit's programming parameters to the parameters stored in your computer. You do not need to wait for this verification process to complete in order to start programming your RTU. If for any reason you wish to cancel the verification process, click on Diagnostics and select "Show Register Verification." (Figure 2a) The Abort button will appear on the lower left side of the form.

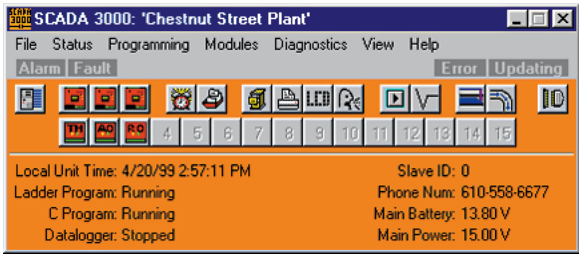

SCADA 3000: 'Chestnut Street Plant'
File Status Programming Modules Diagnostics View Help
Alarm Fault
Error Updating
TH AO RO 4 5 6 7 8 9 10 11 12 13 14 15
Local Unit Time: 4/20/99 2:57:11 PM
Ladder Program: Running
C Program: Running
Datalogger: Stopped
Slave ID: 0
Phone Num: 610-558-6677
Main Battery: 13.80 V
Main Power: 15.00 V

Figure 1b: Main programming form without verification selected

From the main menu, select *Program>System*, or click on the System button from the toolbar.

SYSTEM INFORMATION

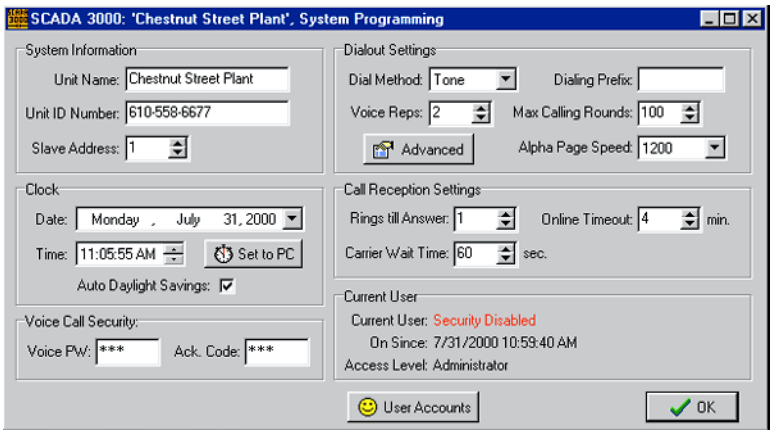


Figure 2: System form

The System form includes the global system parameters that apply to the unit in general.

Unit Name

The Unit Name is the text description of the SCADA 3000. It can be up to 24 characters long. This is automatically filled in from the information you provided when setting up the unit on your computer. When SCADA 3000 sends a cover page with a Fax transmission or alerts an alphanumeric pager, the Unit Description is sent as part of the basic information.

Unit ID Number

The Unit ID Number is the identification number of the SCADA 3000 and can be up to 16 digits long. This number should be programmed to be the same as the telephone number where the unit is installed. The Unit ID Number serves several purposes:

1. It's the first thing spoken by SCADA 3000 during any voice call: "Hello, this is (Unit ID Number)."

2. When SCADA 3000 sends a cover page with a Fax transmission or alerts an alphanumeric pager, the Unit ID Number is sent as part of the information.

Slave Address

This is the modbus address of the SCADA 3000 unit. It is set by default and should not be changed unless you are using multiple units on a radio network. On a radio network each unit must have a unique slave address. The Slave ID can be set to any number between 1 and 247.

CLOCK

Unit Date & Time

The date and time must be programmed into the unit for it to function properly. The time should reflect the actual time where the unit is geographically installed (eg., PST, CST, EST). A button labeled "Set to PC" is located next to the time box. Clicking this button will set the date and time to the present values on your computer.

Auto Daylight Savings

This instructs SCADA 3000 to automatically correct the time twice a year for daylight savings.

Programmable settings: On, Off

Default setting: On

DIALOUT SETTINGS

Dialing Method

The Dialing Method parameter lets you program whether SCADA 3000 will dial out in Pulse or Tone. The default setting is Tone.

Dialing Prefix

The "Dialing prefix" box on the System Settings form is for email prefixes *only*. If for instance the SCADA 3000 is on an office phone system where it had to dial a "9" to get an outside line, entering a "9" here will force the SCADA 3000 to dial "9" before dialing the email server phone number.

To enter a specific prefix for dialout types other than email, go instead to the Dialout Destination screen (see "Destinations" section later in this chapter) and enter the prefix with the phone number in the Destination column, following the instructions under "Special Dialing Codes." You must enter each prefix manually for each individual phone number that requires one.

Maximum Calling Rounds

The maximum number of times SCADA 3000 will dial through the list of destinations to attempt to deliver either an alarm or a report.

Programmable range: 0 - 100 calling rounds
Default setting: 100 calling rounds

Alpha Pager Speed

This is the baud rate of the data connection between SCADA 3000 and your alphanumeric pager service.

Programmable settings: 300, 1200, 2400
Default setting: 1200

Voice Repetitions

The number of times SCADA 3000 repeats the alarm message during a dial out alarm call.

Programmable range: 0-10
Default setting: 2

Advanced Button

Clicking on this button brings up the "Adavanced Dialout" form. The form handles three user-settings.

PC Dialout is for use on SCADA 3000 units connected to radio networks. Checking this box allows the connected PC to handle the Alarm Dialout calling. You must have both a speaker-phone modem and sound card installed in your PC for you to use the PC alarm dialout feature.

Alarm Dialing Priority: When checked this allows the SCADA 3000 unit to override any current PC communications in order to make phone calls when an alarm is triggered.

Email Delivery Settings: The radio buttons here let you choose between the default phone number or a different number for Emails. See *Chapter 21: Email Server Option* for more information.

CALL RECEPTION SETTINGS

Rings Until Answer

This is the number of times SCADA 3000 will let the phone ring before it answers an incoming call. Note that the unit will not answer a call if someone is on-line through one of the RS232 ports.

Programmable range: 1 - 20 rings

Default setting: 1 ring

Carrier Wait Time

The Carrier Wait Time is the amount of time SCADA 3000 will wait for a modem connection when it receives an incoming call.

NOTE: Do not set this parameter too short, otherwise a modem connection may never be established.

Programmable range: 0 to 90 seconds

Default setting: 60 seconds

Online Timeout

The online timeout will force the SCADA 3000 to disconnect from an on-line session if no data requests have been made for the programmed time period. Each time data is requested, the online timeout will reset. If the time period is set to 0, the online timeout feature becomes disabled. This feature prevents the unit from staying connected in the event of a loss in communications.

Programmable range: 0-255 minutes

Default: 4 minutes

Voice Call Security

Asterisks indicate that Voice Password and Acknowledgment Codes are entered. Default values for each are "555." The voice password must be enabled on the voice programming screen. The voice password prevents unauthorized access into voice programming mode. (For information on changing and programming passwords and codes, see "Chapter 9: Communicating with the SCADA 3000.")

Current User Information

This area of the System Form reflects the current user information, including name, access level, how long the user has been logged on, and number of logins. You can change User information here by clicking on the *Change* button, and also check to have the SCADA 3000 remember your password. (For detailed instructions, see the sections on setting up user accounts and changing passwords in Chapter 9.)

UNIVERSAL ADDRESS FORMAT (UAF)

Overview

The Universal Address Format (UAF) is the generic method of referencing almost any parameter inside the SCADA 3000. It is used to configure the Local Display, the Data Logger, the Voice Status Report, the Gas Flow equations, the PID loops, and for accessing information in both the Ladder program and the C Program. In most cases you can simply choose the UAF

from a menu. This section will help you understand how UAFs are defined and provide examples on how to specify them.

The Universal Address breaks down a memory location into three parameters that between them specify where a particular value is stored. The parameters that make up the UAF are called the SOURCE TYPE, GROUP, and ELEMENT.

The SOURCE TYPE refers to the category or type of parameter and is represented by a letter in the UAF. Note that the Main Board has a fixed Module Address of 0 and that the I/O Expansion modules have programmable addresses from 1 to 15. Listed below is a chart that shows how UAFs are constructed.

SOURCE TYPE	GROUP	ELEMENT
I: Input	0-15 (Module Address)	0-15 (Input Channel)
O: Output	0-15 (Module Address)	0-7 (Output Channel)
T: Timer	0-63 (Counter Number)	0-4 (Timer Parameter)
N: Counter	0-63 (Counter Number)	0-6 (Counter Parameter)
B: Bit	0-255 (Word Number)	0-15 (Bit Number)
V: Ladder Variable	0-127 (Variable Number)	0
C: C Variable	0-1023 (C Variable Number)	0
G: Gas Flow	0-3 (Gas Flow Channel)	0-19 (Parameter Number)
A: Alarm	0-63 (Alarm Number)	0
D: Diagnostic	0-15 (Module Address)	0-255 (Diagnostic Parameter)

HOW TO SPECIFY AND APPLY UAFS

Input

An Input would be any physical input channel on either the main board or an input expansion module. The Group for this type would be the module address in the network from 0-15. The Element is the channel within that module. Note that all input channels are numbered starting at 0 not 1.

Examples: I,0,0 = Input 0 on the Main Board
I,5,3 = Input 3 on module 5

Output

An Output would be any physical output channel on either the main board or an output expansion module. The Group for this type would be the module address in the network from 0-15. The Element is the channel within that module. Note that all output channels are numbered starting at 0 not 1.

Examples: O,0,7 = Output 7 on the Main Board
O,2,4 = Output 4 on module 2

Timer

The SCADA 3000 contains 64 timers for use within the Ladder program. The Group for this type is the timer number from 0-63. The Element specifies an individual function for the selected timer. There are 5 elements associated with each timer:

Element Number	Element Description	Range of Value
0	Timer Enabled	0 or 1 (0=disabled, 1=enabled)
1	Timer Running	0 or 1 (0=disabled, 1=enabled)
2	Timer Done	0 or 1 (0=disabled, 1=enabled)
3	Preset Value	0 to 32,767
4	Timer Value	0 to 32,767

Examples: T,0,0 = This UAF indicates if Timer 0 is enabled or disabled

T,12,4 = This UAF references the current timer value of timer number 12

Counter

The SCADA 3000 contains 64 counters for use within the Ladder program. There are two types of Counters, Count Up and Count Down. The Group for this type is the counter number from 0-63. The Element specifies an individual function for the selected counter. There are 7 elements associated with each counter:

Element Number	Element Description	Range of Values
0	Up Counter Enabled	0 or 1 (0=disabled, 1=enabled)
1	Down Counter Enabled	0 or 1 (0=disabled, 1=enabled)
2	Counter Done	0 or 1 (0=disabled, 1=enabled)
3	Counter Overflow	0 or 1 (0=disabled, 1=enabled)
4	Counter Underflow	0 or 1 (0=disabled, 1=enabled)
5	Counter Preset	-32,768 to 32,767
6	Counter Value	-32,768 to 32,767

Examples: C,0,0 =This UAF indicates if Counter 0 is enabled or disabled

C,19,6 = This UAF references the current value of Counter number 19

Bit

The SCADA 3000 contains 4096 bits, arranged as 256 16-bit words, that can be used within the Ladder program. Bits can only have one of two possible values, 0 or 1. The Group for this type refers to the word number from 0 to 255. The Element refers to the actual bit number from 0 to 15.

Word	Bit Number															
Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1
:																

:

Examples: B,0,0 = This UAF refers to bit number 0 of word number 0
B,2,12 = This UAF refers to bit number 12 of word number 3

Ladder Variable

The SCADA 3000 contains 128 user-defined floating point variables for use within the Ladder program. The Group specifies which of the 128 variables is referenced. The Element field is not used and should be set to 0.

Examples: V,0,0 = This UAF references ladder variable number 0
V,35,0 = This UAF references ladder variable number 35

C Variable

The SCADA 3000 allows up to 1024 floating point variables to be defined in the C program. The Group specifies which of the 1024 variables is referenced. The Element field is not used and should be set to 0.

Examples: C,0,0 = This UAF references C variable number 0
V,64,0 = This UAF references C variable number 64

Gas Flow

The SCADA 3000 calculates gas flow for up to four meter runs. The following UAFs refer to various Gas Flow parameters and computed results. The Group refers to one of the four meter runs 0-3. The Element refers to the specific parameter within the specified meter run. A list of the Elements follows:

Element	Description
0	Gas flow calculation Enable/disable (0=disabled, 1=enabled)
1	Gas density (specific gravity)
2	Carbon dioxide content (mole %)
3	Nitrogen content (mole %)
4	Dynamic viscosity (lbs-mass/foot-sec)
5	Btu factor (BTUs/MCF)
6	Real-time MCF per day (updated once per minute)
7	Volume for today (MCF)
8	Volume for yesterday (MCF)
9	Volume for this month (MCF)
10	Volume for last month (MCF)
11	Volume for the hour (MCF)
12	Average differential pressure for the hour (inches of H2O)
13	Average temperature for the hour (deg F)
14	Average static pressure (PSI)
15	Total BTUs for the hour (MBTUs)
16	Flowing minutes for the hour

17	the hour from 0-23, over which the most recent hourly calculation was run
18	Energy for yesterday (BTUs)
19	Energy for last month (BTUs)

Example: G,0,6=real-time flow rate for meter run 0.

Alarms

The SCADA 3000 has 64 programmable Alarms. The Alarm UAF will return a number from 0-3 that will identify if an alarm exists or not and also if the alarm is a High-level or Low-level alarm (High-level and Low-level refer to analog signals that moved above or below the programmed alarm limits). The Group specifies which of the 64 Alarms is referenced. The Element field is not used and should be set to 0. The table below shows the value that will be returned for each alarm condition:

Value	Description
0	No Alarm
1	General Alarm (digital or closure)
2	Low-level Alarm
3	High-level Alarm

Examples: A,0,0 = This UAF references the alarm status for Alarm 0

A,19,0 = This UAF references the alarm status for Alarm 19

Diagnostics

The SCADA 3000 Main Unit and expansion modules maintain a set of data for diagnostic evaluation. The Group number refers to the module address of the hardware in question. A Group number of 0 identifies the Main Unit and Group numbers 1-15 correspond to module addresses 1-15. The Element specifies each individual diagnostic parameter. The Element parameters are different for the Main Unit and the modules. Listed below are the Element numbers and descriptions for the Main Unit.

Element	Description
0	Month (1-12) from the internal clock
1	Day (1-31) from the internal clock
2	Year (1999-2xxx) from the internal clock
3	Hours (0-23) from the internal clock
4	Minutes (0-59) from the internal clock
5	Seconds (0-59) from the internal clock
6	Reference Voltage for Inputs (5V)
7	Reference Voltage for audio (2.5V)
8	Lithium battery voltage (3V)
9	External 12V Battery Voltage (13.8V)
10	Main Power Supply Voltage (15V)
11	Internal 15V Power Supply (15V)

12	Internal 24V Power Supply (24V)
13	SCADA 3000 circuit board temperature in deg C
14	Total Current Consumption in Amps
15	Communication processor flash memory version
16	Control processor flash memory version
17	PCB revision
18	Communication processor status
19	Control processor status
20	Watchdog counter
21	Communication processor reset count
22	Control processor reset count
23	Control Switch Position
24	Day of Week
25	Communication Flash Memory Boot Version
26	Control Flash Memory Boot Version

Expansion Module Diagnostics

Element	Description
0	Module type
1	Module firmware version
2	Module firmware Month
3	Module firmware Day
4	Module firmware Year
5	Module processor stack usage
6	Module communications total packet time-outs

I/O PROGRAMMING

The I/O (Input/Output) programming forms allow you to configure your inputs and outputs so that they accurately describe the signals you will be measuring and controlling. Individual Input and Output names can be programmed in addition to units of measure. Custom tables may be configured for linear 0-5 volt and 4-20 mA transducers. Outputs can be forced On or Off as needed in both manual mode or automatically via Ladder or C-program.

To program I/O information, either click the *Programming* menu and choose *I/O Points*. Or you can click on one of the three Main Unit I/O Programming buttons—the first two for inputs and the latter for Outputs.

The following form will appear:

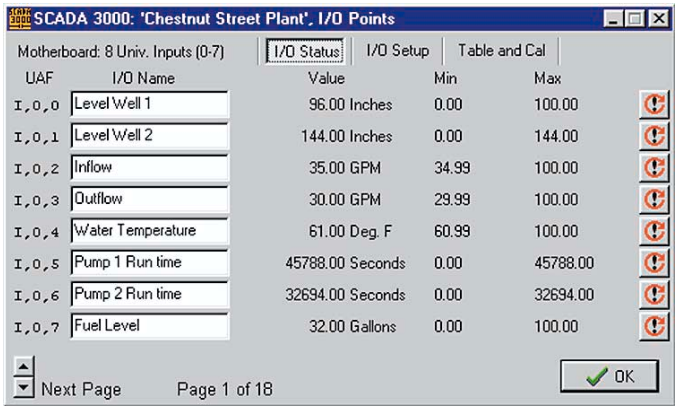

SCADA 3000: 'Chestnut Street Plant', I/O Points
Motherboard: 8 Univ. Inputs (0-7)
I/O Status
I/O Setup
Table and Cal
UAF
I/O Name
Value
Min
Max
I,0,0 Level Well 1 96.00 Inches 0.00 100.00
I,0,1 Level Well 2 144.00 Inches 0.00 144.00
I,0,2 Inflow 35.00 GPM 34.99 100.00
I,0,3 Outflow 30.00 GPM 29.99 100.00
I,0,4 Water Temperature 61.00 Deg. F 60.99 100.00
I,0,5 Pump 1 Run time 45788.00 Seconds 0.00 45788.00
I,0,6 Pump 2 Run time 32694.00 Seconds 0.00 32694.00
I,0,7 Fuel Level 32.00 Gallons 0.00 100.00
Next Page
Page 1 of 18
OK

Figure 3: I/O Points form

The first page is the programming form for inputs 0 through 7 on the main unit, corresponding to the first of the three Main Unit I/O buttons. Using the arrows in the bottom left corner of the form, you can move to the next page(s). Each input can have its own name programmed to distinguish it. The name can be up to 24 characters. This screen will show the current value, the chosen units of measure, and the minimum/maximum values for each input. Min and Max in this case reflect the minimum and maximum values that the input channel has attained. On the far right of each input channel listed is a Reset button. If you click the Reset button, it will set the Min and Max to the Current Value. So if temperature is currently 60ºF, and you click the Reset, both Min and Max will show 60 as their values. As the temperature changes, Min or Max will adjust accordingly. Note: If the input channel is set as a Runtime counter, clicking the Reset button for that input channel will clear Min, Max AND the current Runtime count, and the Runtime will start again from 0.

Click on the I/O Setup button to move to the second screen and configure the input type and units of measure for each input. The I/O setup screen follows.

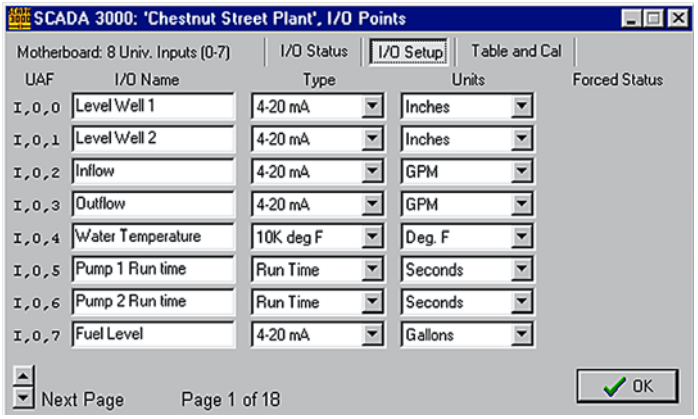

SCADA 3000: 'Chestnut Street Plant', I/O Points
Motherboard: 8 Univ. Inputs (0-7)
I/O Status
I/O Setup
Table and Cal
UAF
I/O Name
Type
Units
Forced Status
I,0,0 Level Well 1 4-20 mA Inches
I,0,1 Level Well 2 4-20 mA Inches
I,0,2 Inflow 4-20 mA GPM
I,0,3 Outflow 4-20 mA GPM
I,0,4 Water Temperature 10K deg F Deg. F
I,0,5 Pump 1 Run time Run Time Seconds
I,0,6 Pump 2 Run time Run Time Seconds
I,0,7 Fuel Level 4-20 mA Gallons
Next Page
Page 1 of 18
OK

Figure 4: I/O setup form

INPUT TYPE

There are six different choices for Input Type. Select the one that matches the transducer or signal source for your application:

- *Closure:* This sets the input to look for a dry contact closure either Normally Open or Normally closed. No external voltage should be applied. When the input jumper is configured for a contact, a low-current 5V sense voltage is applied at the input terminals.
- *Run Time:* This sets the input to start a timer whenever a closed contact is applied to the input. The timer will stop when the contact opens. The input will accumulate the time until the reset button is clicked or the time overflows the maximum. Run time is displayed as *hours:minutes:seconds*. When referencing a Run Time input from a C or Ladder program the value will be provided in seconds (eg. 05:12:45 will be 18765 seconds).
- *0-5 volt:* This sets the input to read the voltage present at the input. The input range of the SCADA 3000 is 5 volts. You can set engineering units on the Table and Calibration tab to scale the 0-5V signal linearly.
- *4-20 mA:* This sets the input to read the current flowing into the positive input. You can set engineering units on the Table and Calibration tab to scale the 4-20mA signal linearly. SCADA 3000 presents a 237 Ohm load when configured for 4-20mA.
- *10K deg F:* This sets the input to read temperature in degrees Fahrenheit from a 10K thermistor. There are several 10K thermistor standards available on the market. Make sure you select one that matches the thermistor table in Appendix C.
- *10K deg C:* This sets the input to read temperature in degrees Celsius from a 10K thermistor. There are several 10K thermistor standards available on the market. Make sure you select one that matches the thermistor table in Appendix C.

UNITS

You can select the units of measure for your input by clicking on the down arrow in the units box for each input. When you select one of the choices from the list, SCADA 3000 will speak the units during status and alarm reports. For contact Closure input types there are dual units of measure to represent the open/closed status of the input. For these input types the first word is associated with an Open contact, the second with a Closed contact. For example, if the selected units of measure is *Stopped/Running*, when the contact is Open SCADA 3000 will indicate *Stopped* and when the contact is closed it will indicate *Running*.

You may also specify your own custom units by selecting *Custom* at the end of the list. This can be any word up to 16 characters. The units will be displayed on the screen; however, they will

not be spoken during a status or alarm report. For contact closure inputs you can specify a word to represent the Open condition and one for the Closed condition by placing a slash between the words. For example, *Cold/Hot* would indicate *Cold* for an Open contact and *Hot* for a closed contact. See below:

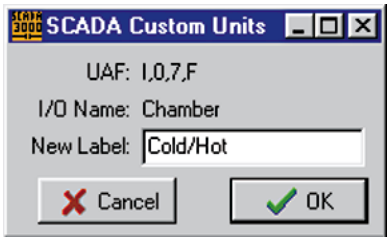


Figure 5: Custom Units

TABLES AND CALIBRATION

To program the custom table values and calibration, click the *Table and Cal* button. The following screen will appear.

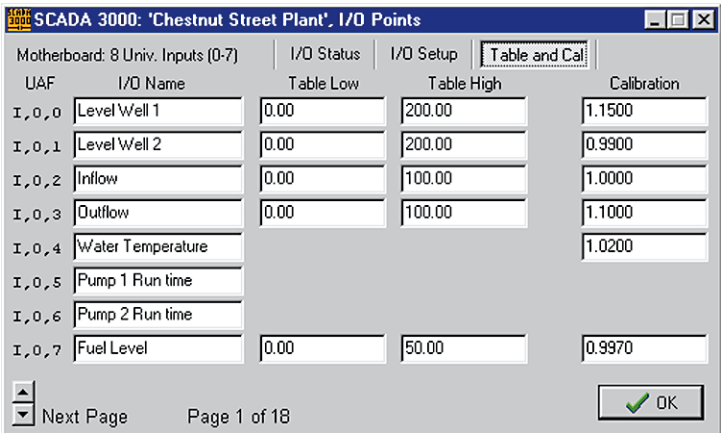


Figure 6: I/O table form

You can set up custom tables for 0-5 volt and 4-20 mA input types so that the values are read in engineering units. For example, say you have a 4-20 mA transducer connected to a 400 inch tank and the transducer puts out 4 mA when the tank is empty and 20 mA when the tank is full. If you enter a table low value of 0 and a table high value of 400, the reading displayed by the SCADA 3000 will be in inches, indicating the amount of material in the tank.

The calibration factor is a multiplier used to correct errors in your system. When set to 1.000 the calibration factor has no effect on the reading. To compensate for an input that is reading high, set the calibration just below 1.0, to a number like 0.98. To compensate for an input that is reading too low, try entering 1.02 for the calibration factor. You can continue to adjust the calibration by trial and error until the reading is accurate or you can compute the percent error directly and enter the calibration factor accordingly.

UAF

Inputs and Outputs can be referenced within the SCADA 3000 using the individual channel's physical address. The physical address is referred to as the UAF, or Universal Address Format. The identifying UAF for each channel is listed on the left column of all screens on the I/O form. The UAF format is used extensively in the SCADA 3000 for reading, writing, referencing or assigning parameters. They are required when programming Alarms, Voice messages,

Datalogging, PID control, C-programs, Ladder programs, and Gas Flow. See "Universal Address Format" earlier in this chapter for more detailed information on the UAF format.

OUTPUTS

The SCADA 3000 has 8 relay outputs on the main board which can be controlled either manually or automatically via C or Ladder programs. To access the Outputs programming form, click on the Motherboard Outputs button from the toolbar. The following form will appear:

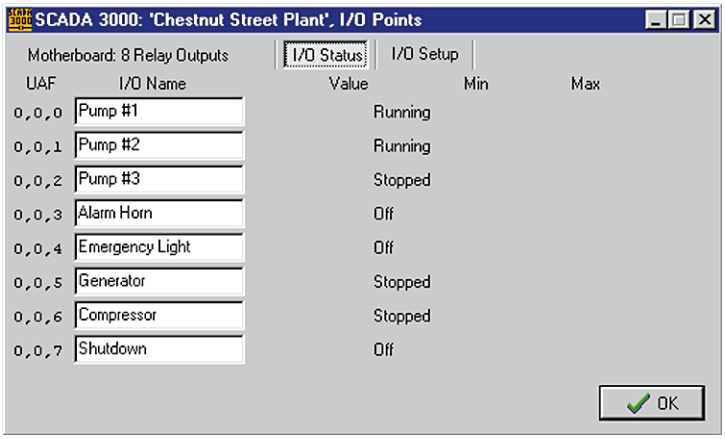


Figure 7: I/O Points Status screen (outputs)

This form allows you to enter a name (up to 24 characters) to describe the device that the output controls. It's best to be as specific as possible when naming the outputs. If you click on the I/O Setup tab, the next page of programming will appear:

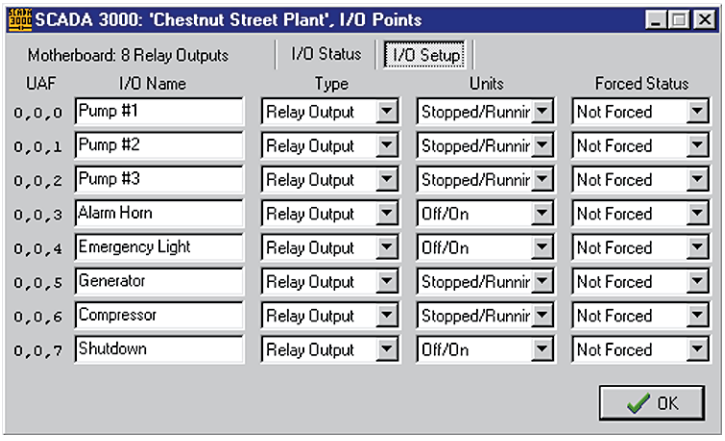

SCADA 3000: 'Chestnut Street Plant', I/O Points
Motherboard: 8 Relay Outputs
I/O Status
I/O Setup
UAF
I/O Name
Type
Units
Forced Status
0,0,0 Pump #1 Relay Output Stopped/Runnir Not Forced
0,0,1 Pump #2 Relay Output Stopped/Runnir Not Forced
0,0,2 Pump #3 Relay Output Stopped/Runnir Not Forced
0,0,3 Alarm Horn Relay Output Off/On Not Forced
0,0,4 Emergency Light Relay Output Off/On Not Forced
0,0,5 Generator Relay Output Stopped/Runnir Not Forced
0,0,6 Compressor Relay Output Stopped/Runnir Not Forced
0,0,7 Shutdown Relay Output Off/On Not Forced
OK

Figure 8: I/O Points Setup screen (outputs)

On this form, the output type is just for reference and is not programmable. You can select Units to describe how the Output device operates when the relay output is Open or Closed. The first word applies to an Open relay the second to a closed relay. The default setting for units is *Off/On*, corresponding to Open relay = Off and closed relay = On. Selections made from the list of Units will be spoken during status reports.

You may also specify your own custom units by selecting *Custom* at the end of the list. This can be any two words up to 8 characters each. The first word applies to an Open relay the second to a Closed relay. The units will be displayed on the screen however they will not be spoken during a status or alarm report. For example, *Disabled/Enabled* would indicate *Disabled* for an *Open*

contact and *Enabled* for a *Closed* contact. See following figure:

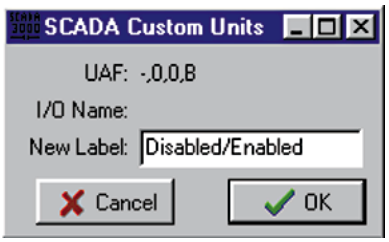


Figure 9: Custom Units (Output)

The last column is labeled Forced Status. The Forced Status gives you the ability to set the output for automatic control or manual control. The three choices are Not Forced, On or Off. When set to Not Forced, the output is controlled automatically from either the C or Ladder program. The other two options, On (closed) and Off (open), will manually force the output to these states. When set to On or Off, the outputs cannot be controlled automatically and can only be changed from the software.

UAF

Inputs and Outputs can be referenced within the SCADA 3000 using the individual channel's physical address. The physical address is referred to as the UAF, or Universal Address Format. The identifying UAF for each channel is listed on the left column of all screens on the I/O form. The UAF format is used extensively in the SCADA 3000 for reading, writing, referencing or assigning parameters. They are required when programming Alarms, Voice messages, Datalogging, PID control, C-programs, Ladder programs, and Gas Flow. See "Universal Address Format" earlier in this chapter for more detailed information on the UAF format.

Chapter 11: PROGRAMMING FOR PHONE LINE APPLICATIONS

DESTINATIONS

SCADA 3000 will call up to 64 telephone numbers or E-mail addresses (both referred to as destinations) to report alarm conditions or send reports. Destinations can be voice calls, fax machines, computers with modems running SCADA 3000 software, E-mail addresses, numeric pagers, or alphanumeric pagers. Choose Program from the main menu, then select Destinations. You can also access the Destination programming from the Toolbar button labeled *Destinations*.

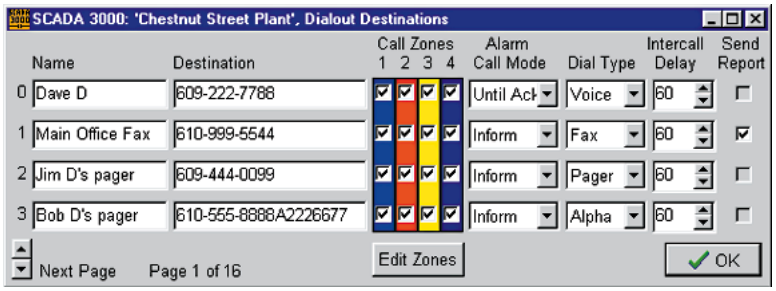


Figure 15: Dialout Destinations form

The Dialout Destinations form allows you to program information for up to 64 destinations. The parameters and their options follow:

NAME

This field allows you to program a description (16 characters max) for each destination. These descriptions are used to reference the destinations in other parts of the programming. The name will also appear on faxes and E-mail transmissions whenever an alarm or report is sent to the destination. If you are entering a fax destination be sure to use the name of the person who you intend to receive the fax.

DESTINATION

The actual telephone number or E-mail address SCADA 3000 dials to deliver its report or alarm message. The Destination field can be up to 64 characters long for E-mail addresses and 50 digits for all other Dial Types and may consist of numbers, letters, and special dialing codes.

Numeric Pagers

The SCADA 3000 can send alarm messages to numeric pagers. It will automatically send its telephone number and the alarm number when dialing to a numeric pager. All you have to do is enter the telephone number of your pager.

Example: (301)565-2300

Note: If your pager service is answered by a voice-prompted message, or if your pager service allows only a few seconds to enter a number before disconnecting, you will need to enter additional programming. See the Special Dialing codes section below.

Alphanumeric Pagers

The SCADA 3000 can dial alphanumeric pagers to send alarm messages. The SCADA 3000 will send the unit description, telephone number, input name and the real-time input value. To

program an alphanumeric pager destination, enter the pager service data or modem phone number followed by the letter A and then the pager ID.

Example: 1-610-555-4593 A 0504099

E-mail

The SCADA 3000 can send alarm messages and reports via e-mail. In the "Destination" field, enter the e-mail address. For this to work you must have an internet e-mail account and register the unit with Phonetics. The unit will automatically dial a toll-free number to the e-mail server at Phonetics and then forward the information to you. A monthly fee is required to receive e-mail from your SCADA 3000. Simply fill out the e-mail registration card that came with your unit, or contact Phonetics for more details. Other e-mail delivery options may be available. Contact factory for details.

Example: Jhalio@AOL.com

Special Dialing Codes

Sometimes, when dialing out on certain phone systems, SCADA 3000 may need special instructions to access an outside phone line, to contact numeric pagers, or to use alphanumeric pager services. The following dialing codes give SCADA 3000 instructions on how to send the phone digits when dialing the phone number. Each code is counted as one digit toward the total of 50 digits. The dialing codes are:

P = two-second pause. A two-second pause can be placed anywhere within the phone number by typing the letter P (upper or lower case). The pause takes up one digit and may be used more than once. This is typically required when a pager service is answered by a voice prompted message, or when a pager service allows only a few seconds to enter a number before disconnecting. You must add enough pauses to get past the voice message. In this case you must enter the number you wish to be displayed on your pager.

Example: 555-2233 PP 444-8877#

POUND (#) or ASTERISK (*): When dialing to a numeric pager, a pound sign (#) or an asterisk (*) may be used within the phone number. An asterisk (*) typically is displayed as a dash (-) on most pagers.

Example: 16105554591 PP 986*0331#

W = wait for answer. This code instructs SCADA 3000 to wait until the call is answered before continuing. The "W" code takes up one digit and may be used only once. The "W" code is typically used in the middle of a dialout number when calling a numeric pager. The SCADA 3000 will wait for the pager company to answer the call before sending the number to be displayed on the pager.

Example: 1-610-555-4593 W 6105582000

(pager company phone number) W (number to display on numeric pager)

Note that digits following "W" will always be dialed as touch-tones when calling a numeric pager.

I= send alarm number. This code instructs SCADA 3000 to send the alarm numbers that have an alarm to a numeric pager. The "I" code takes up one digit and may be used only once. Note that this is only required if you must insert additional dialing codes (such as pauses) to make your pager work; otherwise the alarm number will be sent automatically.

Example: 555-2233 PP 4448877*I#

Special Alphanumeric Pager Dialing Codes

A = alphanumeric pager ID (required)

C = alphanumeric character password (optional)

A = alphanumeric pager ID: The "A" dialing code is used ONLY for alphanumeric pager destinations. It is placed between the pager company's telephone number and a particular pager's ID number by typing the letter A (upper or lower case). It instructs SCADA 3000 to wait until a connection is made to the alphanumeric pager service's computer system before transmitting the ID number for a particular pager.

Example: 1-610-555-4593 A 0504099

C = alphanumeric character password: The "C" dialing code is used ONLY for alphanumeric pager destinations and only if your particular pager system requires a character password. This password may be required for SCADA 3000 to access the pager service's computer system. If required, it is placed between the pager company's telephone number and the required character password in the dialout sequence by typing the letter C (upper or lower case). The character password must precede the pager ID number.

Example if character password required: 1-610-555-4593 C 000000 A 0504099

(pager company phone number) C (character password) A (pager ID number)

The alphanumeric character password is optional and is, in general, reserved for future services. Length of the password, when used, may be different in some systems. Consult your pager service for more information.

CALL ZONES

SCADA 3000 is capable of assigning different destinations to specific calling groups (zones) to accommodate work shift schedules. There are 4 Call Zones to choose from. To include a destination in a Call Zone, check the appropriate box. To set up Call Zones, click on the "Edit Zones" button and the following form will appear:

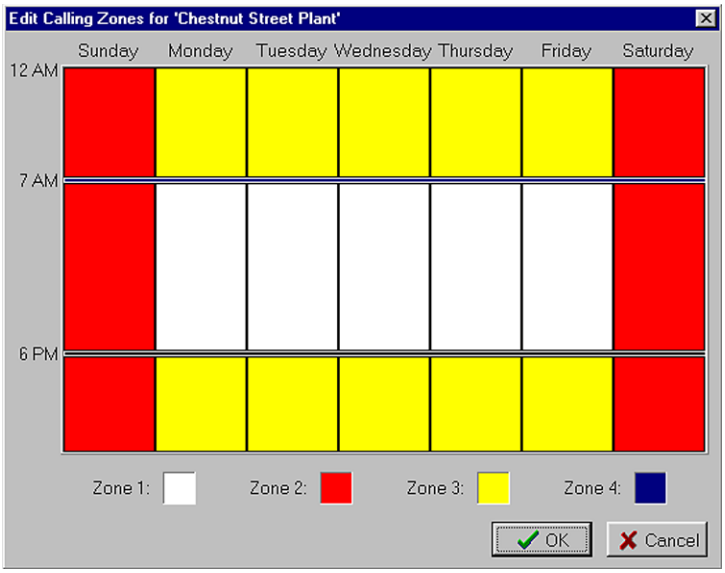


Figure 16: Edit Zones form

The Edit Calling Zones form divides up the 7 day week into 21 blocks of time. To change the start and end time for each block, drag the horizontal divider bars up or down as needed. To

reassign a block of time to a different Call Zone, click inside the block until it matches the color of the desired zone. To change the color of a zone, double click on the zone's color square at the bottom of the form. Select a new color from the palette, then click the OK button.

The Zone configuration in figure 16, creates the following time slots:

Zone 1: Monday-Friday 7:00AM through 6:00PM

Zone 2: Saturday-Sunday 12:00AM through 11:59PM

Zone 3: Monday-Friday 12:00AM through 7:00AM and 6:00PM through 11:59PM

Zone 4: Undefined

In this example you could assign daytime emergency personnel to Zone 1, personnel on call during night time hours (Mon-Fri) to Zone 2, and personnel on call on the weekend to Zone 3.

ALARM CALL MODE

The Alarm Call Mode determines the calling priority for each destination. The three choices are Disabled, Until Acknowledged, and Inform.

Disabled

Destinations programmed with Alarm Call Mode Disabled will never be called. This is simply a way to disable dialing to this destination if necessary. Note that this only affects alarm dialout. SCADA 3000 will still dial to deliver reports if selected.

Until Acknowledged

Destinations programmed with Alarm Call Mode "Until Acknowledged" will continue to be called until one of them acknowledges the alarm. Once acknowledged, the remaining destinations programmed as "Until Acknowledged" will not be called.

> Example: An alarm occurs and the SCADA 3000 is programmed to call destinations #1,#2, and #3 and all 3 destinations are Alarm Call Mode "Until Acknowledged." The unit will call destination #1, if the alarm isn't acknowledged, it will call #2. If destination #2 doesn't acknowledge, then the unit will call destination #3. If the alarm still hasn't been acknowledged, SCADA 3000 will call destination #1 again. This process will continue until one of the destinations acknowledges the alarm.

Note: Fax, E-mail and Modem destinations cannot be programmed as "Until Acknowledged" call modes.

Inform

Destinations programmed with Alarm Call Mode "Inform" will be called until the alarm message has been successfully delivered one time to all "Inform" destinations, regardless of acknowledgment. If a transmission to an "Inform" destination was successful, that destination will not be contacted again for that alarm. SCADA 3000 will only make additional attempts to contact an Alarm Call Mode "Inform" destination if that destination hasn't successfully received its transmission (for example, due to a "no answer," "busy signal," "no carrier" response). If a transmission fails to reach a destination after the last calling round is completed, the failure is recorded in the Event Log. Calls to "Inform" destinations are useful for providing records of alarm events.

> Example: The SCADA 3000 used in the above example is monitoring the heating system in a school. The first three destinations are set as Alarm Call Mode "Until Acknowledged" and programmed to call maintenance personnel. In addition, destination #4 is programmed to call the school's fax machine and is set as Alarm Call Mode "Inform." SCADA 3000 will call the maintenance personnel as described

above so that they can fix the boiler, and will also send a fax to the school office informing them of the problem.

Important: Destinations programmed as Alarm Call Mode "Inform" cannot acknowledge alarms.

DIAL TYPE

This is used to specify the type of call SCADA 3000 will make to a particular destination. The available dialout types are: Fax, Voice, E-mail, Pager, Alpha Pager, and Modem.

Fax - This instructs SCADA 3000 to send a fax transmission to either a fax machine or a PC with the ability to receive a fax directly.

E-mail - This instructs SCADA 3000 to deliver a message to the E-mail address programmed in the destination field.

> NOTE: For "E-Mail" Dial Types, *only* E-mail addresses may be programmed in the destination field, NOT telephone numbers. SCADA 3000 is pre-programmed with a toll-free 800 number that will automatically connect to Phonetics' E-mail server system. E-mail service requires a subscription fee. Contact Phonetics for other e-mail options.

Numeric Pager - This instructs SCADA 3000 to dial a numeric pager and leave the unit's ID number and alarm input numbers on the display. This will happen automatically just by entering the pager's telephone number in the destination field. If you want to change the information that is sent to your pager or you are having trouble receiving the page, you may use the special dialing codes to customize the dialout to your pager.

> NOTE: When dialing to a numeric pager it is sometimes necessary to combine codes. Certain pager systems vary and you must adjust accordingly. To test your pager system, use an extension telephone on the same line as the SCADA 3000 unit and listen in during the SCADA 3000's dial-out to confirm that your pager service is reached without a problem. If you must add a pause, use the letter P to insert a two second pause wherever necessary.

Alpha Pager - This instructs SCADA 3000 to deliver a text message to an alphanumeric pager.

Modem - This instructs SCADA 3000 to call a modem attached to a computer running SCADA 3000 software in "Alarm Receive" mode.

INTERCALL DELAY

The intercall delay is the amount of time SCADA 3000 will wait after completing a call before calling the next destination. Typically, you would want this to be a very short amount of time; however, if a destination is a pager, you may want to set this to 5-10 minutes to allow time to call back after receiving the page.

SEND REPORT

SCADA 3000 can be programmed to automatically send reports to selected destinations. Check this box to include this destination when sending reports.

> Note: Reports may only be sent to Fax or E-mail destinations (See "Reports" later in this chapter.)

VOICE PROGRAMMING

The SCADA 3000 voice option provides the ability to record customized status and alarm messages for playback during status reports and alarm calls to voice phone numbers. The option includes a 90 second pre-recorded vocabulary (referred to as *canned* messages) and a 270 second user-recordable memory. Voice messages can be used to identify alarm conditions, report input/output values, report the results of calculations stored as a variables, or report any parameter accessible through the UAF format: Inputs, Outputs, Timers, Counters, Bits, Ladder Variables, C Variables, Gas Flow, and Diagnostics. Such custom messages are referred to as user-assignable messages. Messages to be used for alarm purposes should be programmed under the *Alarms* form. Messages used for status purposes only should be programmed under the *Voice Setup* form.

> **Note**: User-recorded messages can serve both as Status and Alarm messages. For example, if you recorded the words "Gas Flow Pipeline number one" as a message, this would work to identify that input both for status and alarm situations.

Depending upon your needs, the SCADA 3000 Voice option allows you to split the recordable memory into either 32 or 64 messages. In 32 message mode you have14 seconds to program an Identification (ID) message and 7 seconds for each user-assignable message. In 64 message mode you get 7 seconds to program an Identification (ID) message and 3.7 seconds for each user-assignable message. Voice messages may be recorded remotely, by calling in to the SCADA 3000 with a telephone, or directly, by plugging a telephone into the VOICE PORT jack on the bottom of the unit.

RECORDING AND PROGRAMMING MESSAGES

You can begin recording status and alarm messages at any time. However, we recommend that you first fill out a list of messages, print that list, and then use it as a reference guide for recording. In this way, you won't have to grapple with playing through a list of 64 messages later in order to find which one represents a particular Data Point.

From the *Programming* menu, select *Voice*. The *Voice Setup* form will appear.

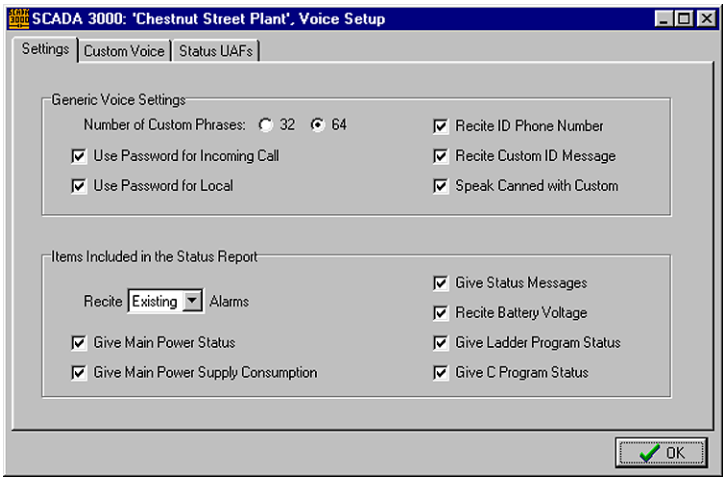


Figure 1: Voice Setup form

The first tabbed screen of this form is the *Settings* screen. Here you select various elements of status and alarm messages, including how many messages you want and whether alarms are recited during status reports. The settings are as follows.

Number of Custom Phrases: Select either 32 or 64. This will determine how many

custom messages you can record and the length of each message (as previously outlined).

Note: Changing from 32 to 64 or vice-versa will erase all previously recorded messages. This is another good reason to make a permanent list.

Use Password for Incoming Call: Checking this box will require you to enter the touch-tone password to obtain access to the voice programming menus when calling-in from a telephone. The default password is set to 555. (See "System Programming" for information on setting the password.)

Use Password for Local: Checking this box will require you to enter the touch-tone password to obtain access to the voice programming menus when connecting a telephone to the Voice Port. Default is 555. (See "System Programming" for information on setting the password.)

Recite ID Phone Number: Checking this box will result in the Unit ID phone number being announced during a status report.

Recite Custom ID Message: Checking this box will result in the custom ID message being announced during a status report. The ID message is message number 64 or 32, depending on the number of Custom Phrases selected.

Speak Canned with Custom: Checking this box will result in the canned vocabulary announcing the input and module number with each custom message.

Examples of what the unit will say with and without canned messages:

a) with canned speech: "Input 3 on module 10 *(well number 1 water level)* is too high—it is now 600 inches."

b) without canned speech: "*(Well number 1 water level)* is too high—it is now 600 inches."

Items Included in the Status Report

Recite Alarms: There are three choices for having alarm information included in the status report. The choices and a description of each are listed below:

No: This setting programs the unit to leave out alarm status information during a status report.

All: This setting will announce the status of all alarms programmed and enabled on the Alarm Programming screen.

Existing: This setting will only announce alarms that currently exist, whether acknowledged or not. This includes any value that **currently** exceeds its specified high or low limits and has met its programmed recognition time.

Recite Status Messages: Checking this box will result in the programmed status UAF messages (see Voice Status Message Programming) being announced during a status report.

Recite Battery Voltage: Checking this box will result in the battery voltage being announced during a status report. Full charge is between 13.5 and 13.8 Volts.

Recite Main Power Status: Checking this box will result in the canned vocabulary announcing if the main power is ON or OFF.

Recite Main Power Supply Consumption: Checking this box will result in the canned vocabulary announcing the present current consumption in Amps.

Recite Ladder Program Status: Checking this box will result in the canned vocabulary announcing the ladder program status, either "running" or "not running."

Recite C Program Status: Checking this box will result in the canned vocabulary announcing the C-Program status, either "running" or "not running."

Sample Status report *(custom messages in parentheses)*

"Hello, this is *(610-555-1234)*"

"*(This is the emergency notification system for the Chestnut Street Water Plant in Little Town, Pennsylvania.)*"

"Alarm channel 20 *(pipeline number 1 inflow pressure)* is too high, it is now 64 psi"

"Input 3 on module 0 *(tank number 3 overflow valve)* is OPEN"

"Input 7 on module 2 *(outflow water temperature)* is now at 85.21 degrees Fahrenheit"

"Relay output 1 on module 4 *(backup power generator)* is OFF"

"Battery voltage is now at 13.72 volts, Okay"

"Main power is ON"

"Power supply consumption is now at 0.45 amps"

"Ladder program is running"

"C-Program is not running"

"Enter Password" (only if you press a touch-tone button)

"Goodbye"

Customizing Your Status Report

Your status report can include information from almost any parameter stored in the SCADA 3000. Besides the general items selected on the Voice Settings form, you can also include information on Inputs, Outputs, Timers, Counters, Bits, Ladder Variables, C Variables, and Gas Flow. You decide which items are reported by selecting them from a master list of all available data points. These items may be reported using the canned message descriptions, or you may assign a custom message to each item.

Return to the Voice Setup form. This time choose the Status UAFs tab.

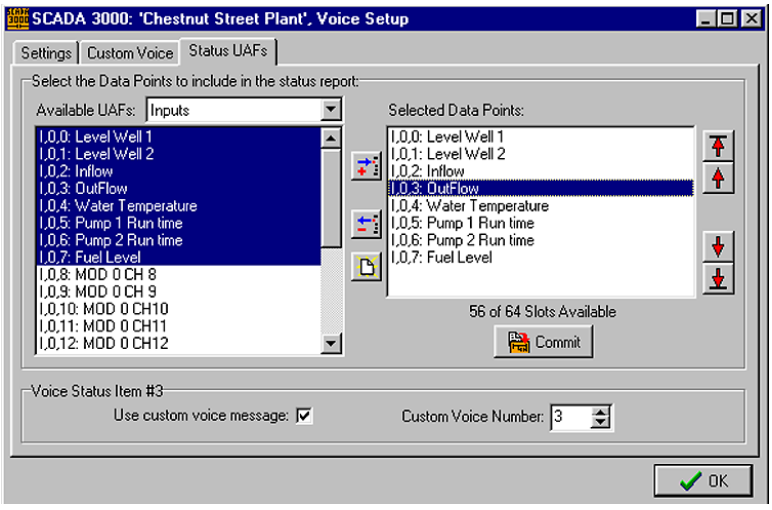


Figure 3: Voice Status UAFs Tab

The box on the left contains a list of all Available data points in a particular category. Use the pop-up menu above it to select the category of data point list you want to see.

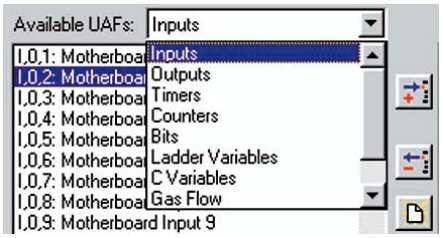


Figure 4: Status UAFs pop-up menu

In each category, highlight the data point(s) you want to use and then click on the top button in the middle column to move your choices to the "Selected data points" list on the right. If you make a mistake, you can highlight the data point you don't want and click the middle button to remove it from the list. If you decide you are not happy with the list, you can erase it completely to start from scratch by clicking the bottom of the three buttons.

Once you have the list assembled, you can rearrange the order of the data points by clicking on the four arrows to the right. The top and bottom arrows will move the selected data points to the top or bottom of the list respectively. The middle two buttons will move them up or down one level.

Linking Custom Voice Messages to Data Points

Click on the first data point in the righthand list and a new area called Voice Status Item, containing two boxes, will appear on the bottom of the Status UAFs screen. To the left is a checkbox marked "Use custom voice message." Check this box if you want to use one of the user-recordable messages as part of the status message for this data point. If you check it, then either scroll through or type in the appropriate number for the "Custom Voice Number" on the right.

Go to the next data point in the list you want to link with a custom message, and repeat the process, until you have set all of your data points and their messages.

When you have the list compiled, click on the Commit button.

Next click on the Custom Voice tab to bring up the Custom Voice Message Information screen.

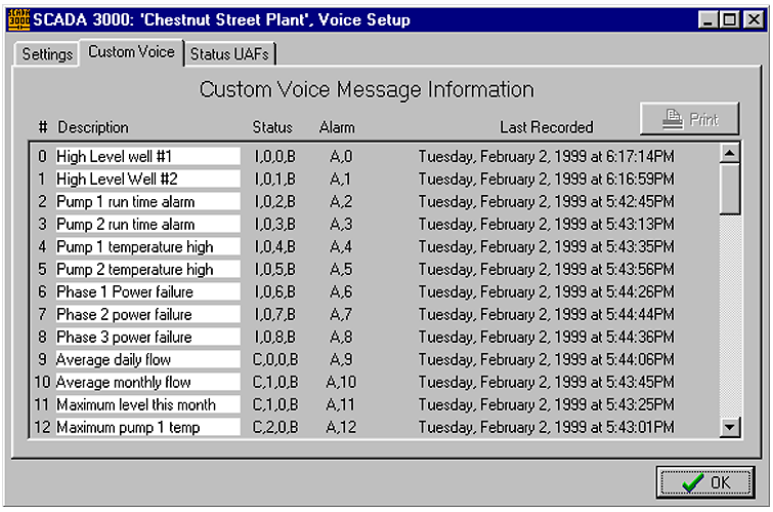

SCADA 3000: 'Chestnut Street Plant', Voice Setup
Settings
Custom Voice
Status UAFs
Custom Voice Message Information
Print
Description Status Alarm Last Recorded
0 High Level well #1 I,0,0,B A,0 Tuesday, February 2, 1999 at 6:17:14PM
1 High Level Well #2 I,0,1,B A,1 Tuesday, February 2, 1999 at 6:16:59PM
2 Pump 1 run time alarm I,0,2,B A,2 Tuesday, February 2, 1999 at 5:42:45PM
3 Pump 2 run time alarm I,0,3,B A,3 Tuesday, February 2, 1999 at 5:43:13PM
4 Pump 1 temperature high I,0,4,B A,4 Tuesday, February 2, 1999 at 5:43:35PM
5 Pump 2 temperature high I,0,5,B A,5 Tuesday, February 2, 1999 at 5:43:56PM
6 Phase 1 Power failure I,0,6,B A,6 Tuesday, February 2, 1999 at 5:44:26PM
7 Phase 2 power failure I,0,7,B A,7 Tuesday, February 2, 1999 at 5:44:44PM
8 Phase 3 power failure I,0,8,B A,8 Tuesday, February 2, 1999 at 5:44:36PM
9 Average daily flow C,0,0,B A,9 Tuesday, February 2, 1999 at 5:44:06PM
10 Average monthly flow C,1,0,B A,10 Tuesday, February 2, 1999 at 5:43:45PM
11 Maximum level this month C,1,0,B A,11 Tuesday, February 2, 1999 at 5:43:25PM
12 Maximum pump 1 temp C,2,0,B A,12 Tuesday, February 2, 1999 at 5:43:01PM
OK

Figure 2: Custom Voice Tab

This list is generated automatically from the custom messages you have assigned on the Status UAFs screen and from the Alarm Notification screen. Up to 64 messages can be listed here, from 0-63.

> **Note**: Message #64 (or message #32 if you choose the 32 messages option) is the recordable ID Message that the SCADA 3000 recites at the beginning of any status report to identify the RTU.

Type in a brief description for each one to identify status/alarm messages in order to help you record them. Thus, Description 0 might read "Refrigeration Unit One," and Description 1 might be "One-hour average temperature", a C-program calculation. Once you've typed in the list of messages in order, you can print this list and use it as your guide for the next step—recording your custom messages.

What you type here does not have to match verbatim what you plan to say—it's merely a means of noting what the message represents.

Once you have your list, then record your custom messages.

Recording and Playback of Messages

Voice messages may be recorded and played back either locally by plugging a telephone directly into the Voice Port jack, or remotely by calling in to the unit. Both methods are outlined below. Note: Pressing the pound (#) key as soon as you have finished speaking a message will end the recording.

Local Voice Programming

Plug the cord of an ordinary analog touch-tone telephone into the Voice Port jack. When you lift the handset the unit will begin reciting a status report.

> **Note**: You must be off-line with the unit in order to record voice messages.

If you've checked the "Use Password for Local" box on the Voice Setup form, a password will be required to access voice programming. Press any key to interrupt the unit if you don't want to hear the status report. It will ask you for the password. The default password is 555. If you wait until the end of the status report, enter a touch-tone to access voice programming. Listed below is the voice menu you will hear:

> "To record message, press 1"
>
> "To play message, press 2"
>
> "To test message usage, press 3"
>
> "To erase message, press 4"
>
> "To hear status report, press 5"

Using your touch-tone keypad, enter the number that corresponds to the function you want to perform. Listed below are the messages that will follow depending on your selection:

1 (Record) "Enter message number, followed by the pound key." The unit will then beep to indicate it is recording. It will beep again when recording is finished.

2 (Play) "Enter message number followed by the pound key." The unit will play the message then beep when the message is over.

3 (Test) "Enter message number followed by the pound key." The unit will play the message in context adding on the appropriate status or alarm information. It will beep when the message is over. The message number must be assigned on the voice or alarm forms for this feature to work.

4 (Erase) "Enter message number followed by the pound key." The unit will beep to indicate that the message was erased.

5 (Status) The unit will play a full call-in status report followed by a beep when the report is over.

The main menu will be recited after each programming option. To end your programming session, simply hang up.

Remote Voice Programming

Voice messages may be recorded or played back remotely by calling in to the unit. You must be off-line with the unit in order to record your voice messages.

When SCADA 3000 answers an incoming call, the unit will beep. Immediately after the beep press any touch-tone key to enter voice mode. If you don't press a key within a few seconds, the unit will try to negotiate a modem connection. If the modem connection fails, the unit will begin to recite a status report. Press any key to cancel the status report and enter the voice programming menu. If you checked "Use Password for Incoming Call," the unit will say "Enter Password" and you must enter it in the Voice Password to access the Voice Programming menu. The default Voice Password is 555. If you wait until the end of the status report, enter a touch-tone to access voice programming. Listed below is the voice menu you will hear:

"To record message, press 1"

"To play message, press 2"

"To test message usage, press 3"

"To erase message, press 4"

"To hear status report, press 5"

"To program ladder variable 127, press 9"

Using your touch-tone keypad, enter the number that corresponds to the function you want to perform. Listed below are the messages that will follow depending on your selection:

1 (Record) "Enter message number, followed by the pound key." The unit will then beep to indicate that the beginning of the recording. It will beep again when recording is finished.

2 (Play) "Enter message number followed by the pound key." The unit will play the message then beep when the message is over.

3 (Test) "Enter message number followed by the pound key." The unit will play the message in context adding on the appropriate status or alarm information. It will beep when the message is over. The message number must be assigned on the voice or alarm forms for this feature to work.

4 (Erase) "Enter message number followed by the pound key." The unit will beep to indicate that the message was erased.

5 (Status) The unit will play a full call-in status report followed by a beep when the report is over.

9 (Ladder) If you press '9' the unit will say "Variable 127 is now at ___, enter number followed by the pound key." You can enter a single digit between 0 and 9 using your touch-tone telephone. The numerical value you enter will be copied into ladder variable 127. This allows you to remotely initiate a control sequence within your ladder program simply by adding statements that check the value of ladder variable 127. Once the desired logic has executed, you can reset ladder variable 127.

The main menu will be recited after each programming option. To end your programming session, simply hang up.

To Hear a Status Report

To hear a status report, call the unit from a touch-tone telephone and press any key immediately after the beep. The unit will begin speaking a status report.

To hear a status report on-site, plug a telephone into the Voice Port jack. The unit will begin speaking a report the moment you lift the handset.

ALARM PROGRAMMING

The SCADA 3000 has the capability to alarm on 64 events. An alarm event can be set from an input, output, variable or any value that has a UAF. Each alarm has a programmable recognition time which must be met before the alarm truly exists. Here are some examples.

Input Alarm: An input (temperature sensor) exceeds its programmed high limit continuously for 2 minutes; however, the recognition time is programmed to be 5 minutes. An alarm will not occur until the condition has persisted for 5 minutes.

Output Alarm: An output (pump) is programmed to set an alarm if it stays on for more than 30 minutes (a 30 minute recognition time). The alarm will not be set unless the pump has been on for 30 continuous minutes. If the pump shuts off after 28 minutes, the recognition time will start over from 0 the next time the pump turns on.

Variable Alarm: A calculation within a C program computes the average volume of material traveling through a pipe. This variable is referenced as the input source for an alarm. The calculation must result in a value that exceeds the programmed alarm limits, for the programmed recognition time, before an alarm will occur.

When an alarm is detected, three things happen: 1)If any of the dialout destinations are programmed and enabled, the unit will begin dialing. 2)A record of the alarm will be entered into the event log. 3)The alarm status will be set to reflect the type of alarm (1 for a general alarm; 2 for a low-level alarm; 3 for a high-level alarm). See Chapter 10 for more detailed information regarding UAFs.

To program alarm information, click on *Programming*>*Alarms*, or click on the *Alarms* button on the toolbar. The alarm status form will appear:

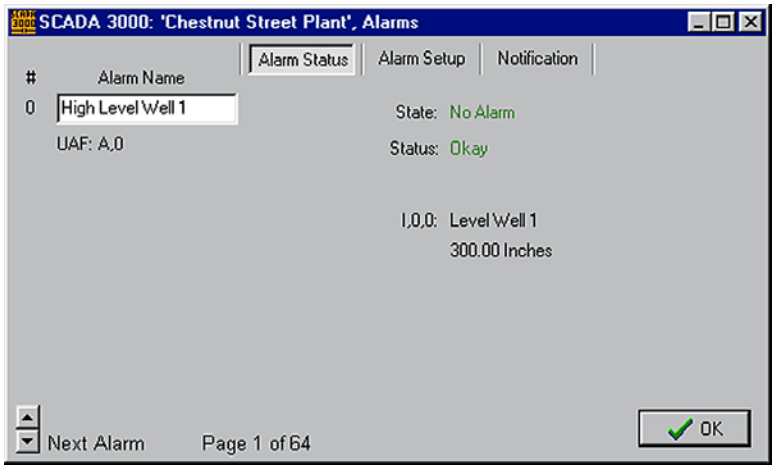


Figure 11: Alarm Status

Alarm programming includes three screens: Status, Setup, and Notification. The Status screen shows the current state of any already programmed alarms. To check on them, use the arrow buttons in the lower left to move through the list.

Alarm Name

Each alarm can have its own name (up to 24 characters) that can be used to describe the alarm. Since alarms can be set from any source, it is important to enter a name and be as specific as possible. If you set an alarm from an input, it is recommended that you use the same name for both the Input Name and the Alarm Name.

State

This parameter describes the state of the alarm. It tells you whether or not the condition has met the recognition time, if the alarm exists, if it has been acknowledged or if it has returned to its normal range. A description of each of the possible messages is listed below:

No Alarm: No alarm exists at the present time. The alarm source is within its programmed limits.

Waiting: The source of the alarm is presently exceeding its limits but hasn't met the recognition time yet.

Alarm, Alarm High, Alarm Low: The source of the alarm has exceeded its alarm limits for the programmed recognition time. If the source is a Normally Open or Normally Closed alarm type, the word *Alarm* will display. If the source is an analog signal the words *Alarm High* or *Low* will display to indicate if the High limit or Low limit has been exceeded.

Cleared: The alarm has been acknowledged but the source is still outside of the programmed alarm limits.

Outstanding**:** The alarm is set but the source has returned to its normal range.

Status

This parameter describes the current status of the source with respect to the source's alarm programming. There are four possible messages:

Okay: The alarm source is within the programmed limits.

High, Low: The source of the alarm is an analog signal it presently is above (high) or below (low) the limit.

Not Okay: The source of the alarm is a digital signal (closure or bit) and is presently in a fault-condition state.

I/O Point Value

This parameter is the actual value of the alarm source as programmed on the alarm setup form.

ALARM SETUP

The Alarm Setup screen is where you configure alarms. Each alarm channel has a programmable source of information that is used to detect an alarm condition. This source of information can be any piece of data that has a physical address defined by the UAF (Universal Address Format). This includes inputs, outputs, variables and other types of data. Assign a corresponding name for each alarm channel. Be sure Alarm Enabled is checked.

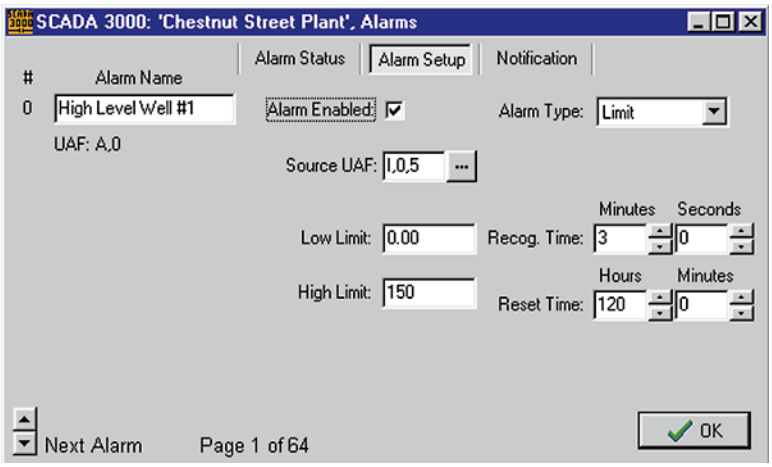


Figure 12: Alarm Setup

A description of each programming parameter on this screen is listed below:

Alarm Enabled: This box can be used to enable or disable alarms before or after they are setup. When this box is not checked (e.g., disabled), the alarm will not be detected.

Source Data Point: This parameter identifies the physical location of the signal to be monitored. It can be any valid data point in the SCADA 3000. The first 16 alarms default to the UAF locations for the 16 inputs on the motherboard. In the dropdown box next to the heading, choose the data point category by clicking on the down arrow. The choices are: inputs, outputs, timers, counters, bits, ladder variables, C variables, gas flow, alarms, and diagnostics. When you pick a category, a list of all possible points in that category will appear. Click on the item you wish to use as the alarm source, and click *OK*.

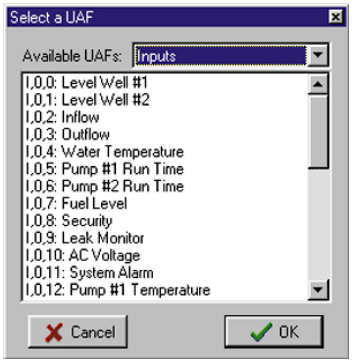


Figure 13: Source Data Points

Alarm Type: This parameter describes the source as being an analog value or a digital value. There are three choices: Normally Open/Bit =0, Normally Closed/Bit=1, and Limit.

If your source is an analog signal (e.g., temperature, or tank level) then choose limit, as this type of signal will involve programming a high and low limit. Limit is for a data point where values may fluctuate but must remain within a specified range.

If your signal is a digital value (open/closed, 1 or 0) then choose the alarm type that reflects the *normal* condition of your source. For example, if your source is a contact closure on an input and it is normally closed (opens upon alarm), then choose Normally Closed/Bit=1. Normally Open triggers an alarm if the circuit closes. Normally Closed triggers an alarm if the circuit opens.

Low/High Limits**:** If your source is analog you must program low and high limits to determine when an alarm condition exists. Low default is 0.0; high is 100.0.

Recognition Time: This is the time required for an out-of-tolerance condition to qualify as an alarm event. The input must remain in that required state according to chosen type continuously for this entire period of time in order to become an alarm. The default is 3 seconds. The programmable range is from 0 minutes and 0 seconds to 1000 minutes and 40 seconds.

Reset Time: This is the time allowed for an acknowledged alarm's fault condition to be corrected before the unit resets (reactivates) the alarm and begins the dialout process all over again. Default of 0 hours and 0 minutes means that the alarm reset feature is disabled. The programmable range is from 0 hours and 0 minutes to 1000 hours and 40 minutes.

> Example: The alarm reset time is set to 3 hours. At 1 a.m., an alarm occurs on input 7. The unit calls Destination 1 and the person who answers the call acknowledges the alarm. Three hours later the fault condition still exists, and the alarm reactivates and begins the dialing sequence again.

ALARM NOTIFICATION

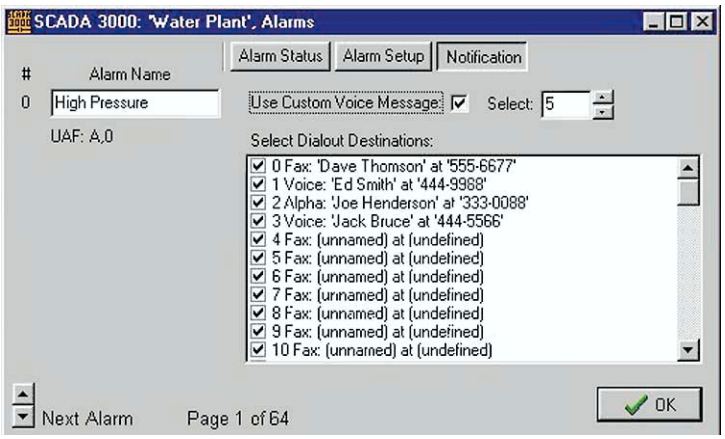


Figure 14: Alarm Notification Screen

On this screen you establish which alarms call which dialout destinations and assign any custom messages you've created into the alarm message that will be spoken. Any of the 64 (or 32) Voice messages you create can be assigned to any of the 64 specific alarms as well as status messages.

Custom Voice Message: Check the box to use a custom message with the selected alarm.

Select: Select the message number to assign to this alarm. You can either type the number of the message or use the up and down arrows to scroll through the list.

Select Dialout Destinations: Click in the small checkbox beside each programmed destination that you wish to have notified should this specific alarm occur.

Repeat this process for each alarm.

When you're finished, click *OK*.

The SCADA 3000 has the capability to alarm on 64 events. An alarm event can be set from an input, output, variable or any value that has a UAF. Each alarm has a programmable recognition time which must be met before the alarm truly exists. In addition, each alarm also has programmable high and low limits, alarm reset time and the option of speaking a custom voice message.

There are three stages to complete an alarm event: 1) Alarm recognition, 2) Alarm notification, and 3) Acknowledgment of the alarm. Note that not all alarm conditions will go through each stage. For example, some may not meet the recognition time.

ALARM RECOGNITION

Each I/O point or value that is monitored for an alarm condition has user-programmed limits to determine when an alarm exists. When the status of this monitored value changes or exceeds the programmed limits, it causes a first-stage alarm condition. At this stage the alarm recognition timer starts running but no dialout occurs. If this condition lasts long enough to meet its programmed recognition time, the alarm becomes active and the SCADA 3000 begins the alarm notification sequence.

ALARM NOTIFICATION

SCADA 3000 can make 6 types of phone calls: Voice, Fax, E-mail, Modem, Numeric Pager, and Alphanumeric pager.

Dialout Note: Call Progress

SCADA 3000 monitors the progress of telephone calls when it is dialing out for alarms. After it dials, it will allow the line to ring up to ten times. If the call is not answered before ten rings have occurred, the unit will hang up, wait the programmed intercall time, and then dial the next number. If SCADA 3000 dials out and encounters a busy signal, it will hang up immediately, wait the programmed intercall time, and then go on to the next number in sequence.

ALARM CALL MODE: UNTIL ACKNOWLEDGED VS. INFORM

Until Acknowledged

Destinations programmed with Alarm Call Mode "Until Acknowledged" will continue to be contacted until one of them acknowledges the alarm.

> Note: Fax, E-mail, and Modem destinations cannot be programmed as "Until Acknowledged."

Inform

Destinations programmed with Alarm Call Mode "Inform" will be contacted until the alarm message has been successfully delivered one time. If a transmission to an "Inform" destination was successful, that destination will not be contacted again.

> Important: Destinations programmed as Alarm Call Mode "Inform" CANNOT ACKNOWLEDGE ALARMS.

Alarm Dialout - Voice

When dialing out to a destination programmed as "voice," SCADA 3000 waits for the phone to be answered, then recites its user-recorded identification message, then the message identifying the value that has gone into alarm.

Below is an example of what SCADA 3000 might say during a typical 'voice' dialout:

> "Hello, this is (ID number)
>
> ("Custom ID message")
>
> "Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.
>
> "Hello, this is (ID number)
>
> ("Custom ID message")
>
> "Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.
>
> "Hello, this is (ID number)
>
> ("Custom ID message")
>
> "Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.
>
> "Enter acknowledgment code."

In this example, the number of Voice Message Repetitions was set to *three*.

> Note: If the call reaches an answering machine, the message will be recorded, but SCADA 3000 will be talking over your outgoing message, so you will probably lose part of the first alarm message repetition.

Alarm Dialout - Numeric Pager

When dialing out to a destination programmed as "Pager," SCADA 3000 leaves a number (usually the unit's own phone number) on the display of a numeric pager.

> IMPORTANT: When dialing out to a phone number programmed as "Pager," SCADA 3000 DOES NOT speak a voice message. It calls the pager company or service, enters the number to be displayed on the pager, then hangs up.

Alarm Dialout - Alphanumeric Pager

When dialing out to a destination programmed as "Alpha," SCADA 3000 leaves a text message up to 80 characters long on the display of an alphanumeric pager.

> IMPORTANT: When dialing out to a phone number programmed as "Alpha," SCADA 3000 DOES NOT speak a voice message. It calls the pager company or service, transmits the information to be displayed on the pager, then hangs up.

Alarm Dialout - Modem

When dialing out to a destination programmed as "Modem," SCADA 3000 waits for the phone to be answered by a modem, establishes a data connection, then transmits its alarm message.

> IMPORTANT: When dialing out to a phone number programmed as "Modem," SCADA 3000 DOES NOT speak a voice message. It is expecting the call to be answered by a PC running SCADA 3000 Windows software. Once a connection is established, the SCADA 3000 transmits the alarm information, then hangs up.

Alarm Dialout - Fax

When dialing out to a destination programmed as "Fax" SCADA 3000 waits for the phone to be answered by a fax machine, establishes a connection, then faxes its alarm message.

> IMPORTANT: When dialing out to a phone number programmed as "Fax," SCADA 3000 DOES NOT speak a voice message. It is expecting the call to be answered by a fax machine. Once a connection is established, the SCADA 3000 faxes the alarm information, then hangs up.

Alarm Dialout - E-mail

When dialing out to a destination programmed as "E-mail," SCADA 3000 is pre-programmed to call the toll-free 800 number of the Phonetics' E-mail server system. Once connected, the unit delivers its alarm message to the E-mail address programmed in the destination field.

> IMPORTANT: When dialing out to a phone number programmed as E-mail, SCADA 3000 DOES NOT speak a voice message. It is expecting the call to be answered by the Phonetics' E-mail server system. Once a connection is established, the SCADA 3000 E-mails the alarm message, then hangs up.

ALARM ACKNOWLEDGMENT

Alarm Acknowledgment - Voice Dialout

Repeated below is the same example of what SCADA 3000 might say during a typical "voice" dialout:

> "Hello, this is (ID number)
>
> ("Custom ID message")

"Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

"Hello, this is (ID number)

("Custom ID message")

"Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

"Hello, this is (ID number)

("Custom ID message")

"Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

"Enter acknowledgment code."

SCADA 3000 will now wait 5 seconds for the Touch-Tone acknowledgment code to be entered. After the last digit of the acknowledgment code has been received, SCADA 3000 will respond by saying: "Alarm Acknowledged by (last number dialed)." The alarm has been acknowledged and the unit will hang up. Once the alarm has been acknowledged, the dialout process stops.

If the Touch-Tone acknowledgment code is not received, the SCADA 3000 will offer you a second chance to enter it, responding with: "beep," "error," "Enter acknowledgment code." If the acknowledgement code is still not received, then the SCADA 3000 will respond by saying: "beep," "error," "goodbye." The alarm has not been acknowledged. SCADA 3000 will hang up and wait for a callback acknowledgment. This waiting period is called the "intercall delay time." During this time you may call the unit back from a Touch-Tone phone. Wait for the unit to beep. Then hit any key on the phone, and the unit will give a voice report. Once you receive the complete report, enter the code to acknowledge the alarm. NOTE: An alarm cannot be acknowledged using a pulse (rotary) telephone.

Alarm Acknowledgment - Pager Dialout

SCADA 3000 will dial out to your pager service and leave a number on the display of your pager. (See Programming Section) The unit will then hang up without speaking a voice message and wait for callback acknowledgment. This waiting period is called the "intercall delay time." During this time you may call the unit back from a Touch-Tone phone to receive a report of the alarm condition and acknowledge the alarm by entering the acknowledgment code.

NOTE: An alarm cannot be acknowledged using a pulse (rotary) telephone.

Below is an example of what SCADA 3000 will say when you call it back to acknowledge a typical alarm:

"Hello, this is (ID number)

("Custom ID message")

"Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

"Hello, this is (ID number)

("Custom ID message")

"Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

"Hello, this is (ID number)

("Custom ID message")

"Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

"Enter acknowledgment code."

SCADA 3000 will now wait 5 seconds for the Touch-Tone acknowledgment code to be entered. After the last digit of the acknowledgment code has been received, SCADA 3000 will respond by

saying: “Alarm Acknowledged.” The alarm has been acknowledged and the unit will hang up. Once the alarm has been acknowledged, the dialout process stops.

If the Touch-Tone acknowledgment code is not received, SCADA 3000 will offer you a second chance to enter it, responding with: “beep,” “error,” “Enter acknowledgment code.” If the acknowledgement code is still not received, then the SCADA 3000 will respond by saying: “beep,” “error,” “goodbye.” The alarm has not been acknowledged. SCADA 3000 will hang up and wait for a callback acknowledgment. This waiting period is called the “intercall delay time.” During this time you may call the unit back from a Touch-Tone phone. Wait for the unit to beep. Then hit any key on the phone, and the unit will give a voice report. Once you receive the complete report, enter the code to acknowledge the alarm. NOTE: An alarm cannot be acknowledged using a pulse (rotary) telephone.

> TIP: When SCADA 3000 is programmed to make calls to pagers, make sure the intercall delay time is long enough to give the person carrying the pager some time to get to a phone to call the unit back.

Alarm Acknowledgment - Alphanumeric Pager Dialout

SCADA 3000 will dial out to your alphanumeric pager service and leave a message on the display of your pager. (See Chapter 10: Programming, for dialing to pagers). The unit will then hang up without speaking a voice message and wait for callback acknowledgment. This waiting period is called the “intercall delay time.” During this time you may call the unit back from a Touch-Tone phone to receive a report of the alarm condition and acknowledge the alarm by entering the acknowledgment code.

NOTE: An alarm cannot be acknowledged using a pulse (rotary) telephone.

Below is an example of what SCADA 3000 will say when you call it back to acknowledge a typical alarm:

“Hello, this is (ID number)

(“Custom ID message”)

“Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

“Hello, this is (ID number)

(“Custom ID message”)

“Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

“Hello, this is (ID number)

(“Custom ID message”)

“Alarm Channel 1 is too high. It is now 82.90 degrees Fahrenheit.

“Enter acknowledgment code.”

SCADA 3000 will now wait 5 seconds for the Touch-Tone acknowledgment code to be entered. After the last digit of the acknowledgment code has been received, SCADA 3000 will respond by saying, “Alarm Acknowledged.” The alarm has been acknowledged and the unit will hang up. Once the alarm has been acknowledged, the dialout process stops.

If the Touch-Tone acknowledgment code is not received, SCADA 3000 will respond by saying: “beep,” “error,” “goodbye.” The alarm has not been acknowledged. SCADA 3000 will hang up and wait for a callback acknowledgment. This waiting period is called the “intercall delay time.” During this time you may call the unit back from a Touch-Tone phone. Wait for the unit to beep. Then hit any key on the phone, and the unit will give a voice report. Once you receive the complete report, enter the code to acknowledge the alarm. NOTE: An alarm cannot be acknowledged using a pulse (rotary) telephone.

TIP: When SCADA 3000 is programmed to make calls to alphanumeric pagers, make sure the intercall delay time is long enough to give the person carrying the pager some time to get to a phone to call the unit back.

Alarm Acknowledgment - Modem Dialout

SCADA 3000 will dial to a PC running the SCADA 3000 software in Receive Alarm mode and deliver an alarm message that will be logged by the software.

SCADA 3000 will transmit the alarm information to the PC, hang up and wait for a callback acknowledgment. This waiting period is called the "intercall delay time." During this time you may call the unit back from a Touch-Tone™ phone and enter the code to acknowledge the alarm. NOTE: An alarm cannot be acknowledged using a pulse (rotary) telephone.

Alarm Acknowledgment - Maximum Calling Rounds

The SCADA 3000 will dial all of the destinations repeatedly until the alarm is acknowledged. If the unit reaches the maximum number of calling rounds without being acknowledged, it will stop the dialout and acknowledge itself. When this occurs, a message is recorded in the Event Log.

ALARM DIALING PRIORITY

This feature will make alarm dailout the priority if an online session exists. As a result, when the option is enabled, the unit will drop the online session and begin dialing out when an alarm occurs. This feature is useful if you have a computer online with the unit 24 hours a day, 7 days a week, but if an alarm occurs, you want the alarm phone calls to take priority. To enable this feature click on the *System* button from the toolbar or select *Programming*, then *System* from the main menu. Next, click the *Advanced* button within the *Dialout Settings* group. The following form will appear. Click in the *Alarm Dialing Priority* box to enable this option.

VIEWING CURRENT ALARMS

You can view all current alarms from one or more units that are online by clicking the *Alarms* button from the Sensaphone Control Center toolbar, or by clicking *History*, then *View Alarms* from the main menu. You can also acknowledge alarms from this screen. A sample is shown below.

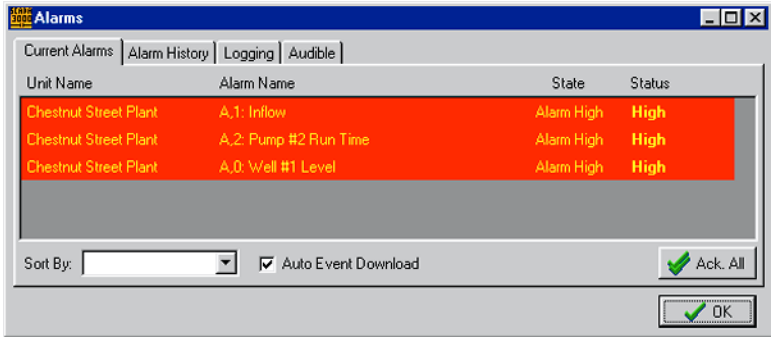


Figure 15: Current Alarms

COMPUTER ALARM RECEPTION

Alarms from your SCADA 3000 unit(s) can be received and logged on a computer when the destination type in your SCADA unit is set to modem. The computer that will be receiving the alarm calls must have a modem, the computer must be on continuously, and it must be running the SCADA 3000 software. To enable your computer to receive alarms click on *Comm* then *Setup* from the Control Center form. Select the *Comm Port* tab which corresponds to your

modem and click the box that says *Answer Incoming Calls*. See sample below. This will set your modem to answer when the phone line rings.

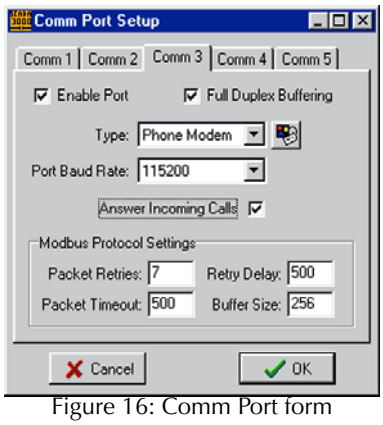


Figure 16: Comm Port form

When an alarm call is received, the software will try to match the calling unit with one in its database. If a match is found, it will download the event log and display the alarm information on the *Alarm History* form shown below.

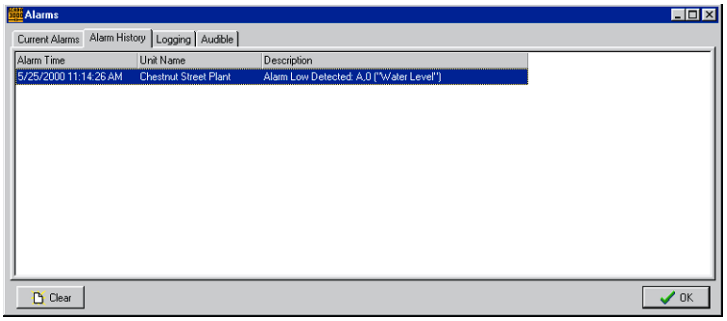


Figure 17: Alarm History form

This form will store all alarms that are received on the computer. You can clear the information on the form by clicking the *Clear* button.

Alarm Printing and Logging

Alarms that are received can also be immediately printed and/or logged to a delimited text file for permanent storage or for importing into other programs. To enable these features click the Alarms button from the toolbar on the Control Center form and then select the Logging Tab. The form below will appear:

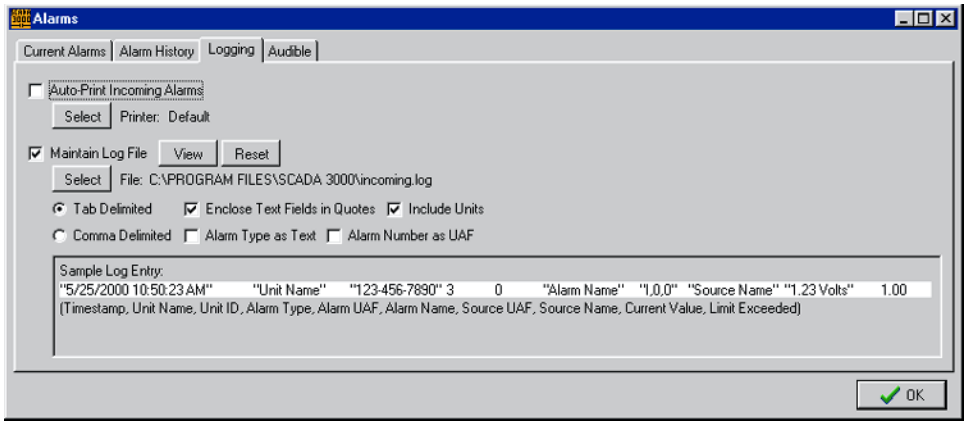


Figure 18: Alarm Log screen

Printing

To print alarms as they are received, click the *Auto-Print Incoming Alarms* box. Next choose a printer by clicking the select button.

Logging

To log alarms to a delimited text file click the box labeled *Maintain Log File*. Next, click the *Select* button to enter a name for the file. The default file name is Incoming.log. Now choose how you would like the fields delimited, either Tab or Comma. Note that Tab delimited files have the option of having the fields enclosed in quotes. Next, choose if you would like the alarm type to be described using text (Alarm, Alarm Low, or Alarm High), by checking the "Alarm Type as Text" box; or check the "Alarm Number as UAF" box to receive it as a number (1,2, or 3) corresponding to the UAF alarm definition. This selects between showing the Alarm Number as a UAF (e.g. A,0) or as a number only (e.g. 0). The log file will be updated each time an alarm is received.

An example of the alarm log entry as you've chosen to format is shown in the "Sample Log Entry" box.

Audible Alarm

The computer receiving the alarm can optionally play an audio message whenever an alarm is received. The computer must have a soundcard and speakers to use this feature. The audio message can be either the Windows default sound or a customized message which can be recorded by the user using standard Windows audio recording tools. The audio file must be in the Windows Wave format. To enable this feature select the Audible tab. Next, click on either the Windows default sound or *Wave File*. To select your own Wave file, click the *Select* button. A default Wave file is included which speaks the message "Sensaphone SCADA 3000 Alarm exists". You can preview the message by clicking the play arrow. The audible programming form is shown below:

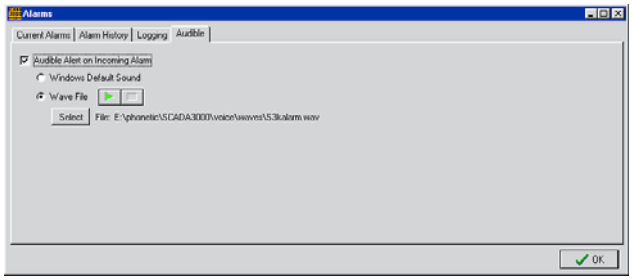


Figure 19: Audible programming form

REPORTS

SCADA 3000 can send periodic reports to selected fax and e-mail destinations (see "Destinations"). Reports may include information from the Event Log and/or the Datalogger. You can select how much information you want to send with each report. To program the report parameters, select *Programming>Reports* from the menu or click on the *Reports* button from the toolbar. The following screen will appear:

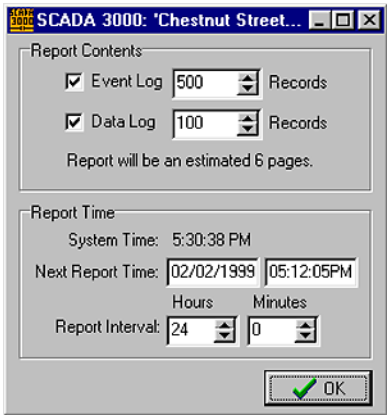


Figure 20: Report form

To receive an Event Log report, click on the Event Log box and enter the number of records you would like to receive. The SCADA 3000's Event Log stores up to 1000 events.

To receive a Data Log report, click on the box and likewise enter the number of records you would like to receive. A Data Log record includes all logged data for one time period.

Set the date and time you would like to receive the next report. This only needs to be set once for the first report. After that, it will automatically send at the programmed intervals. This is useful if you would like your reports sent at the top of the hour or at a specific time of day. The report interval can be set between 0 hours 0 minutes and 199 hours 59 minutes.

DATALOGGING

SCADA 3000's built-in Data Logger can log and store up to 50,000 samples from up to 128 Data Points. All samples are logged at the same time on a programmable time interval. Each stored sample includes a time and date stamp with the year stored as four digits. The Data Log viewer allows you to select or query stored data from multiple units and display it in grid format. From the grid you may copy the data to the Windows Clipboard and paste it into another application such as MS Excel or Lotus 123 or you can export the data in a comma-delimited format. The Data Log Viewer also provides the ability to graph one or more data points on a line chart.

From the main programming form choose *Programming*>*DataLog* or simply click on the *DataLog* button from the toolbar. The Data Log programming screen is shown below:

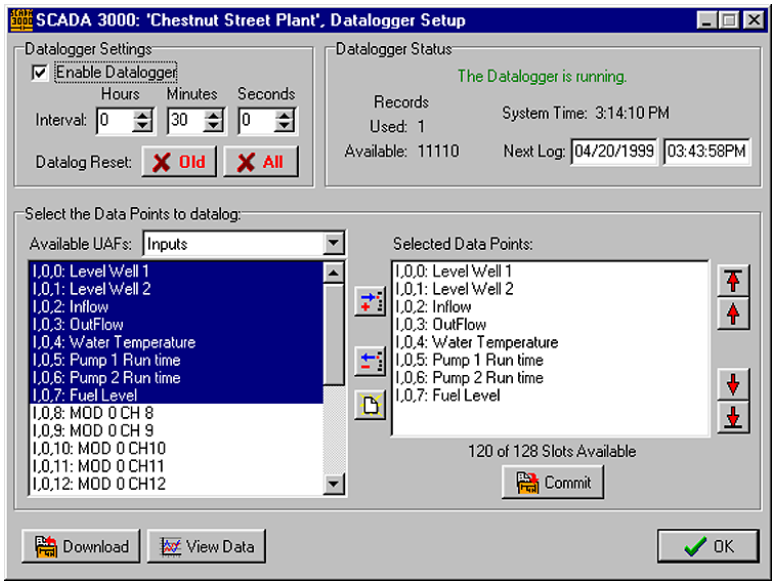


Figure 21: DataLog form

Programming the SCADA 3000 Data Logger requires four steps:

(1) Select the items you wish to log & click on the commit button.
(2) Set the time interval for logging.
(3) Set the time you wish to start the Datalogger.
(4) Click the Enable box.

The following information describes in detail how to complete these steps and also how to retrieve and view the Data Log information.

DATA LOGGING CAPACITY

The SCADA 3000 has enough memory to store up to 100,000 pieces of information, including the time stamp. This means that if you are datalogging one point, the memory would store one time-stamp and one value at each interval, resulting in 50,000 time stamps and 50,000 values—100,000 records. A record thus comprises one time-stamped log of the selected data points. If you select more than one point to be logged, the SCADA 3000 will store a time stamp plus the value for each point selected. The following formula can be used to determine the number of time-stamped records for a given number of points:

$$\text{Time-stamped Records} = \frac{100{,}000}{\#\text{ of points} + 1}$$

As an example, if you wanted to log 35 points, the calculation would compute a capacity of 2,777 records. If your interval was set to 30 minutes, the Data Logger would be able to run for 57.85 days before overwriting any data.

In the Data Logger status window, under the heading *Records*, the unit will always provide you with the number of records used and the number available. You can see the available number change after you select the items to be logged and then click the *Commit* button.

SELECTING DATA POINTS TO LOG

All values in the SCADA 3000 are referenced by the data point. On the form above there are two large list boxes, the one on the left provides a list of the data points that can be chosen for datalogging and the box on the right shows the data points that have been selected for

datalogging. In this process you will be choosing items from the list on the left and copying them to the list on the right. In the drop down box next to the heading *Available UAFs*, choose the type of value you wish to add to the datalogger by clicking on the down arrow. The choices are Inputs, Outputs, Timers, Counters, Bits, Ladder Variables, C Variables, Gas Flow, Alarms, and Diagnostics. When you pick a category, a list of all possible points in that category will appear. Click on the item you wish to add to the Data Log and then click on the first of the three buttons between the two list boxes, it has an arrow pointing towards the right. This will copy the highlighted item to the list of *Selected Data Points* on the right (e.g. those to be Datalogged). Continue this process by highlighting each data point to be datalogged and copying it to the list of *Selected Data Points*. When you are finished, click on the *Commit* button. This will send the list to the SCADA 3000.

To remove a data point from the list, highlight the data point from the *Selected Data Points* list box and click on the middle button, the one with the arrow pointing towards the left. This will remove the highlighted item. When you have finished editing your list, click on the *Commit* button to send the new list to the SCADA 3000.

To clear the entire list of *Selected Data Points*, click on the bottom button. This will delete all data points in the *Selected Data Points* list box.

> **Note**: Changing data points to be logged will delete all presently stored data. Be sure to download the contents of the Data Logger before making any such changes.

ORDER OF DATA POINTS LOGGED

Once you have selected the items to be logged you may want to arrange the data points in a specific order. This may be important because the Datalog viewer will arrange the data in the order it was logged, so if you want to view certain data points side by side you should log them adjacent to each other. To rearrange the *Selected Data Points*, highlight the data point you would like to move, or drag the mouse over a group of data points, and use the arrows to the right to move them up or down. The interior arrows will move the data points up or down one place and the outside arrows will move them to the extreme top or bottom of the list. When finished editing the list, click the *Commit* button.

TIME INTERVAL

The time interval determines how often the SCADA 3000 will log data. The default setting is 1 minute. You may set the time interval to as little as once a second or to several hours between logs. It is recommended that you set this parameter as high as your application allows, for several reasons:

1) Setting the Data Logger to just a few seconds can produce large amounts of data. This will quickly start to consume large amounts of space on your hard drive.

2) A short time interval will cause the Data Logger to fill up faster thus requiring frequent downloads.

3) A full Data Logger will take longer to download.

4) As you begin to acquire more and more data in the datalogger database, the software will start to take longer to perform a query.

DATA LOGGER START TIME

You may program the Data Logger to start at a specific time to keep the data on a fixed time schedule. For example, if you want to log data every minute, you could enter a start time of 12:00:00PM and all of the time stamps would occur on the exact minute, starting at 12:00:00PM. To have the Data Logger start at a specific time, enter the time in the *Next Log* box. Note that

for the start time in the Next Log box to work correctly, the time must be ahead of what the system time indicates. For this reason, the system time is provided just above the *Next Log* programming box. After you have set the time in the *Next Log* box, click the *Enable Data Logger* box. As soon as the *System Time* matches the *Next Log* time, the data Logger will start.

If you do not enter a time in the *Next Log* box you may simply click the *Enable Data Logger* box. This will start the Data Logger immediately.

STOPPING THE DATA LOGGER

To stop the Data Logger click the Enable Data Logger button to remove the check mark. This will stop the Data Logger from writing any new data.

RESETTING THE DATA LOGGER

The Data Logger has two reset options: 1) reset all of the data, and 2) reset only the old data. When you click the Reset *All* button, all of the data in the SCADA 3000 will be deleted and the *Records Used* will return to 0. When you click the Reset *Old* button, only data that has been downloaded previously will be deleted. Therefore any data that has not been downloaded yet will remain in the unit. The *Records Used* parameter will reflect the number of records remaining in the unit. Clicking the Reset *Old* button before downloading the Data Logger can be used to reduce the download time. If multiple users will be downloading data from the Data Logger, care must be taken to delete records ONLY after all users have downloaded or to instruct all users never to delete the Data Logger.

DOWNLOADING THE DATA LOGGER

To get the Data Logger information from the SCADA 3000 to your computer you must download it. This procedure retrieves the data from the SCADA 3000 and stores it in a database on your hard drive. To avoid overwriting data in the SCADA 3000 you must always download the data before the number of available records gets to zero. To download the Data Logger click on the *Download* button on the lower left part of the form. When all of the data has been downloaded the SCADA 3000 software will update the database and provide some statistical information regarding the download process. You may choose to view the data at this time or click OK to return to the datalogging form. The Datalog download form follows:

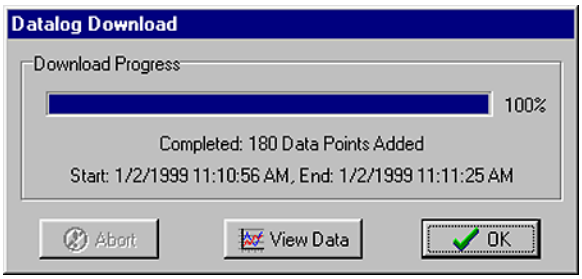


Figure 22: Datalog Download status

VIEWING THE DATA

To view Data Logger data, click on the *View Data* button on the lower left part of the Data Log form or from the Download form. Select one or more SCADA 3000 unit's whose data you want to view. To select more than one unit, drag the mouse over the list or hold the control key down while you click each unit you want to highlight. When finished, click the *Next* button. A list of

all available points, from the units you selected, will be displayed. Choose the data points you want to view by dragging the mouse over the list or hold the control key down while you click each item you want to highlight. Click the *Next* button. Enter the range of dates and times to perform your query and click on the *Query* button. The figure below shows the procedure just described.

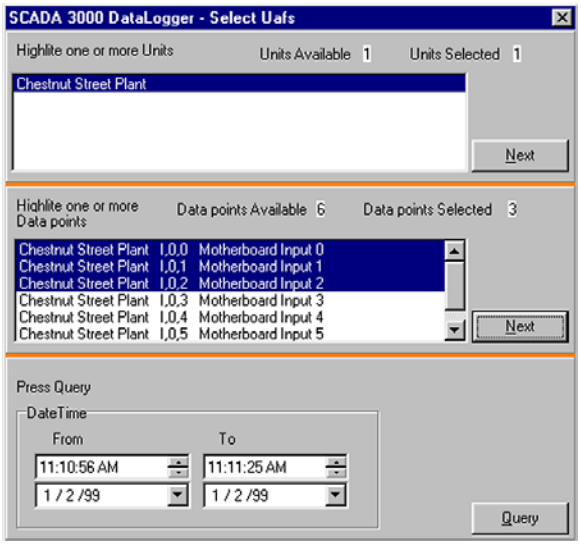


Figure 23: DataLogger Data form

After you click the *Query* button the data which matches your request will be displayed on a grid. Use the up/down arrows to scroll through the data in the grid. At the top of the form are seven buttons which provide additional options that are described below:

Graph

If you click on the graph button, all of the points in the query will be displayed as a line graph. You can use the arrows on the bottom and side of the graph to adjust the position of the graph. The graph may be printed by clicking the *Print* button at the top of the screen. You may also copy the graph to the windows clipboard by clicking the *Copy* button. This allows you to paste the graph into other applications. The *Zoom In* and *Zoom Out* buttons let you view the data with more or less detail. The *Reset* button restores the original view of the graph. When finished, click the *Close* button to return to the Datalog form.

Export

The Export button allows you to save the data in the grid to a comma-delimited file. This data format can then be imported into other applications.

Copy

The Copy button will copy the data in the grid to the Windows Clipboard. This makes it easy to import the data into other applications, such as MS Excel or Lotus 123, because all you have to do is Paste it from the Clipboard.

Delete

The delete button will delete **ALL** of the datalog samples for the data points regardless of the time period selected for the query.

Print

The print button will print all data in the grid.

New Query

The new query button closes the current grid and allows you to build a new query.

Close

Click the close button when you are finished viewing the data. This will return to the DataLog form.

EVENT LOGGER

The SCADA 3000 keeps track of important system events with its built-in Event Logger. Each SCADA 3000 can store up to 1000 system events. System events include user log-ins, alarms, incoming & outgoing telephone calls, and alarm acknowledgments. See Appendix B for a complete list of events stored by the SCADA 3000. The Event Log can be viewed/retrieved several ways: (1) It can be sent as a scheduled report from the SCADA 3000 via fax or e-mail (see Reports); (2) it can be downloaded manually while a user is online; and (3) you can have your computer automatically call one or more units and retrieve the Event Logger based on a polling schedule (see Polling). Once the events have been downloaded to your computer, you can run the Event Log viewer to perform queries on the data. Note that once the event logger in the SCADA 3000 is full, the oldest records will be overwritten by the newest.

DOWNLOADING THE EVENT LOGGER

To download the Event Logger go online with your SCADA 3000 and click on *Status*, then *Download Event Log* from the main menu of the unit programming form. The software will begin downloading the events from the unit and store them in a database on your computer. When this process is finished the following form will appear:

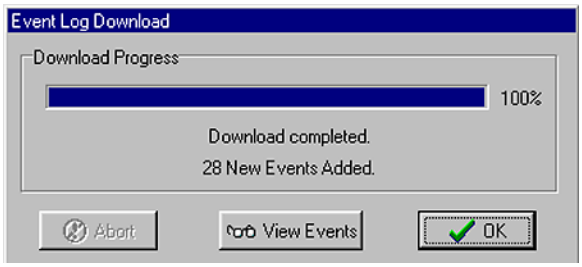


Figure 24: Downloading the Event Logger

To view the Event Log now, click the *View Events* button, or click *OK* if you would prefer not to view them at this time.

VIEWING THE EVENT LOGGER

To view the Event Log click the *View Events* button after downloading or click on *Status*, then *View Event Log* from the main menu of the unit programming form. You may also view events by selecting *History*, then *View Events* from the Sensaphone Control Center form. The Event Log query form will appear. To begin click on *Events*, then select one or more units that you

would like to query. To select multiple units, hold the Control key down and click on the units to include in the query. When you have finished selecting units, click the *Next* button.

Choose the specific events to appear in the query. You may select them from the list or click one of the pre-defined query buttons below the list box. When finished click the *Next* button.

Choose the way the events are ordered by selecting one of the four options in the *Query Order* block. You can view the events in chronological order by newest-to-oldest (Descending), oldest-to-newest (Ascending), alphabetically by event description, or by unit name.

Finally, enter the time period that contains the events you wish to view and click *Query*.

Deleting Events from the Event Log Database

Perform a query as described above and click the *Delete* button to remove the queried events.

Printing, Exporting, and Copying the Event Log

The event log can be printed by performing a query and clicking the *Print* button at the top of the screen. You can also export the Event Log to a comma delimited text file. Simply click the *Export* button after performing a query and enter a filename. Comma-delimited text files can typically be imported into most spreadsheets or databases. As a second option, the Event Log can also be copied to the Windows clipboard. This allows you to simply paste the copied information into other programs that support Windows Cut, Copy, & Paste functions.

POLLING

The SCADA 3000 Software can be programmed to connect with your SCADA 3000 units, one at a time, on a predetermined time schedule, to retrieve the Data Log and Event Log. The downloaded information will be stored in a database in the computer for viewing at a later time. The concept is similar to setting an alarm clock to download a unit's Data Logger and Event Log.

Polling is a useful feature for users who need to save more data than the SCADA 3000 is capable of holding. By polling on an automated, periodic basis, it is possible to log large amounts of data to your computer without overwriting data in the SCADA 3000. It is also a time saver since it can be programmed to perform the data collection in the middle of the night when your computer is typically doing nothing. As an added benefit, telephone rates are typically reduced during these hours. Once the data has been downloaded you can choose to view it using the Data Log or Event Log View options, or export the data to another application for data processing or archiving purposes.

SETTING UP A POLLING SCHEDULE

To set up a polling schedule click on the Polling button from the toolbar of the Control Center screen. The following form will appear:

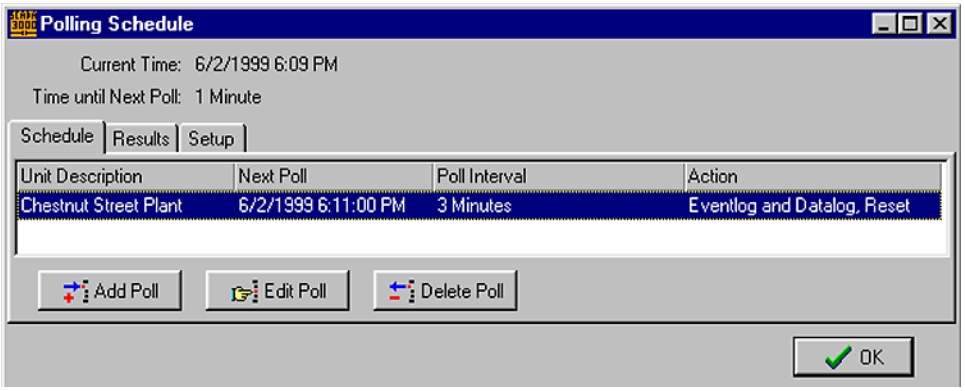


Figure 25: Polling Schedule form

Click on the Add Poll button to add a unit to the polling schedule. The Edit Poll screen will appear as shown below:

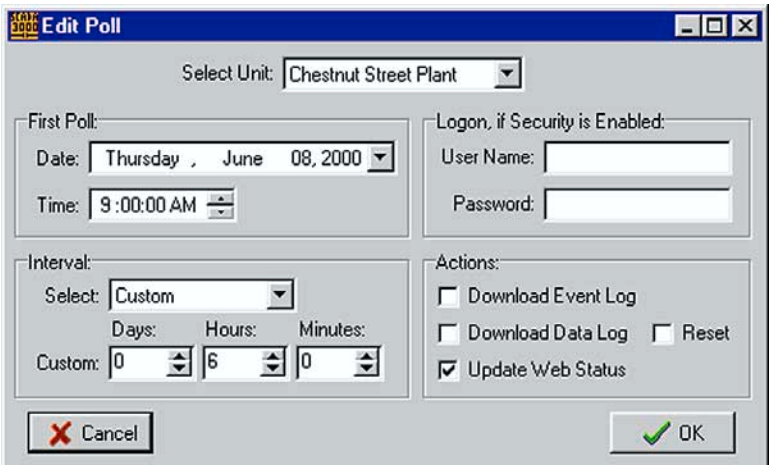


Figure 26: Edit Poll form

Select a unit by clicking the Down arrow in the unit selection box. If security is enabled in the selected unit, enter a username and password to use during the poll. The user must have either Programming or Administrator access to reset the Data Logger. Choose a date and time for the first poll. Next, select a polling interval which will determine how often the software will connect with the selected unit. The polling interval options are: poll once, daily, weekly, monthly, and custom. The options are explained below:

- Poll Once—The software will connect with the selected unit at the time and date specified by the first poll information and then stop.
- Daily—The software will connect with the selected unit the first time at the time and date specified by the first poll information, and then schedule the next poll for the same time the following day.
- Weekly—The software will connect with the selected unit the first time at the time and date specified by the first poll information, and then schedule the next poll for the same day and time the following week.
- Monthly—The software will connect with the selected unit the first time at the time and date specified by the first poll information, and then schedule the next poll for the same date and time the following month.
- Custom—The software will connect with the selected unit the first time at the time and date specified by the first poll information, and then schedule the next poll by

adding the amount of time specified in the custom time field to the time of the previous poll.

After selecting the polling interval, choose which information you would like to download—the Event Log and/or the Data Log. You may also choose to Reset the Data Log after downloading it to minimize the time of the call during the subsequent polls. Note, however, that this will delete information in the SCADA 3000's Data Logger, preventing another user from connecting to the unit to retrieve the Data Log information. When you have finished programming the polling information, click OK.

General Set Up

Next, click on the Setup tab of the Polling Schedule form. The screen shown below will appear:

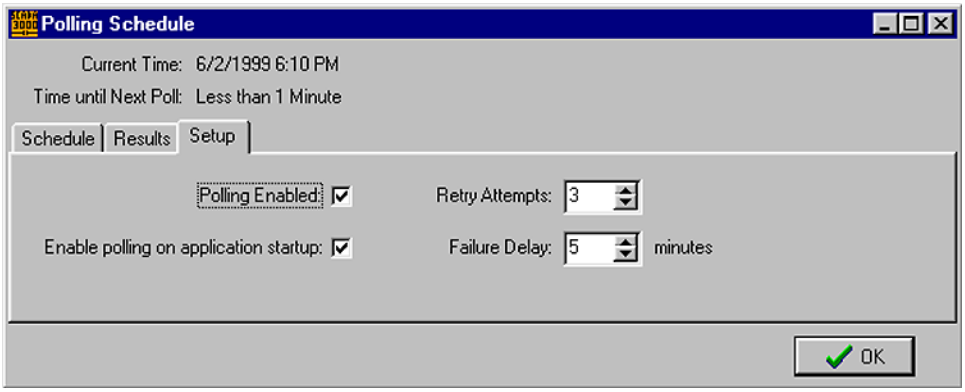


Figure 27: Polling Setup Tab

This screen contains general polling set-up parameters. The Polling Enabled option is, generally speaking, the polling on/off switch. If you enable it, all units in the polling schedule will be called at their respective times. If you disable it, polling is turned off and no units will be called. If you check Enable polling on application startup box, polling will be enabled every time the SCADA 3000 software is run. If you do not check this box, polling will be disabled at startup until you enable it manually. Checking this box is useful if you want your computer to restart the polling process after it has rebooted for some reason such as loss of power or operating system malfunction. You may adjust the number of times the software tries to connect to a unit if it happens to fail the first time. Enter the number in the Retry Attempts box. The default setting is 3. You may also select how long to wait before retrying by entering the time in minutes in the Failure Delay box. The default time is 5 minutes. When you have finished setting these parameters, click OK.

The Results tab on the Polling Schedule form will store general polling related information regarding polling failures, attempts, etc.

CHANGING THE POLLING SCHEDULE

If you wish to change the polling schedule of a particular unit, simply click on the next scheduled poll from the Schedule tab and then click the Edit Poll button. Make your changes, then click OK. The Polling Schedule form always shows the current polling schedule and updates its information after each poll.

Chapter 12: PROGRAMMING FOR RADIO APPLICATIONS

INTRODUCTION

Wireless communication with one or more SCADA 3000s may be accomplished via radio modems. In this scenario, one radio modem would be connected to the host computer and additional radio modems would be connected to each SCADA 3000 on the network.

- The transmit frequency of the host PC radio modem must match the receive frequency of the radio modem at each SCADA 3000. Similarly, the receive frequency of the host PC radio modem must match the transmit frequency of the radio modem at each SCADA 3000.
- The host computer can be used to program an individual unit or to poll each unit on the network to update a real-time screen.
- Each unit on the network must have a unique modbus ID. The host computer will request information from a unit based on its modbus ID. As a result, all of the SCADA 3000 units will "hear" the request for information, but only the unit with a matching modbus ID will actually respond.

SELECTING RADIO MODEMS

There are numerous manufacturers of radio modems, but they all differ slightly. A listing is provided at the end of this section. Some of the options that must be considered when selecting a radio modem include:

licensed/unlicensed operation
baud rate requirements
cost
power requirements
geographical constraints

While most of these issues will depend upon your particular circumstances, we can give you some general information to start off with. Licensed frequency radios are generally cheaper but they require registration with the FCC. Unlicensed radio modems (spread spectrum), while usually more expensive, can be faster to get up and running. Both types of radio modems may be used with the SCADA 3000. The baud rate of the radio modem will determine how quickly information will be transmitted and received. The SCADA 3000's COM2 port can operate over a range of 1200bps to 38.4Kbps. Faster radio modems generally cost more. Consider a radio modem that can communicate at 38.4Kbps for networks with many SCADA 3000 units. Power requirements will vary for different radio modems. Most operate from a 12V supply. Radio modems with greater distance capability typically require more power. Be sure that your power source is capable of powering the radio modem. Geography may determine whether a radio modem can be used at all. Typically, mountainous regions will not work. Line of sight provides the best chances of error-free data transmission.

CONFIGURING SCADA 3000 FOR USE WITH RADIO MODEMS

The SCADA 3000 COM2 port is configured for connection directly to a radio modem. A null modem adapter is **not** required. Set the serial port baud rate and flow control jumper as described in Appendix D. These must be set to match your radio modem. The COM2 serial port may be configured with flow control enabled or disabled. Flow control is enabled when the

COM port configuration jumper is set to the Radio position. In our experience most radio modems do not require flow control; as a result, the COM2 configuration jumper may be left in the STD (standard) position. When flow control is not used, the radio modem will simply continue to send data until a short period of silence occurs.

> Note: Be sure to set each unit's modbus ID to a unique number **BEFORE** powering up all of the units on your network.

FLOW CONTROL

COM port flow control is enabled in the SCADA 3000 when the configuration jumper (See Appendix D) is installed in the Radio position. When flow control is enabled, the unit will wait until Carry Detect is raised before receiving data. The unit will not transmit data until the Carry Detect has been dropped. When the unit wishes to transmit data, it will raise the RTS (request-to-send) line and wait for the CTS (clear-to-send) line to respond. Once this occurs, the unit will send its data and then drop the RTS line. The RTS hold time (hold after data sent) may be adjusted by using the SCADA 3000 Register Map Inspector utility from the *Diagnostics* menu of the SCADA 3000 programming software. Select register 199 "Radio Timeout". The RTS hold time default setting is 00 0A, which corresponds to 100ms. To read the current value of this register, click the *Force Read* button. To change the value, click the *Force Change* button, enter a new code and click OK. A list of valid codes vs. hold times appears below.

Code	Time	Code	Time
00 00 =	0 ms	00 0A =	100ms
00 01 =	10 ms	00 0B =	110ms
00 02 =	20 ms	00 0C =	120ms
00 03 =	30 ms	00 0D =	130ms
00 04 =	40 ms	00 0E =	140ms
00 05 =	50 ms	00 0F =	150ms
00 06 =	60 ms	00 10 =	160ms
00 07 =	70 ms	00 11 =	170ms
00 08 =	80 ms	00 12 =	180ms
00 09 =	90 ms	00 13 =	190ms

HOST COMPUTER SETUP

The radio modem for the host computer connects to an external serial port on the computer. The radio modem for the host computer must have a transmit frequency that matches the receive frequency of the SCADA 3000 radio modems. Similarly, the host computer's radio modem must have a receive frequency that matches the transmit frequency of the SCADA 3000's radio modems. Next, you must configure the serial port settings by running the SCADA 3000 software and selecting *COMM*, then *Setup* from the main menu of the Sensaphone Control Center form. The following form will appear.

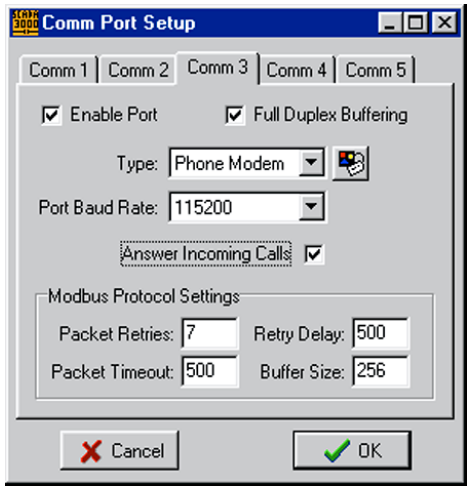


Figure 1: Comm Port Setup form

Select the CommPort tab corresponding to the serial port you will be using to connect your radio modem. You must have Full Duplex Buffering disabled.

Type: Select Radio Modem

Port Baud Rate: set the baud rate to match the communications rate of your radio modem.

Enable Port: Click to enable.

Alarm Dialout: Enables a host PC to perform an alarm dialout when the host computer on the radio network is connected to a phone line.

Adjust the modbus communications (protocol) settings as required. Higher settings may be required for slower networks. Click the button beside the *Type* setting to bring up the following form.

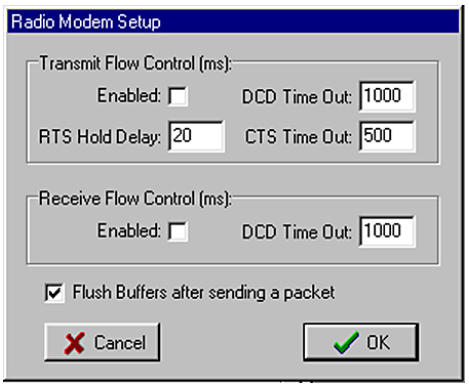


Figure 2: Radio Modem Setup form

These settings are used to program the transmit and receive settings between the host computer and its radio modem. If your radio modem does not use flow control, disable both the Transmit and Receive Flow control boxes shown above. If your radio modem does use flow control, you will need to check Enable for both Transmit and Receive flow control, and adjust the timing parameters to match the characteristics of your radio modem. The following timing diagrams show the logical sequence of the flow control signals.

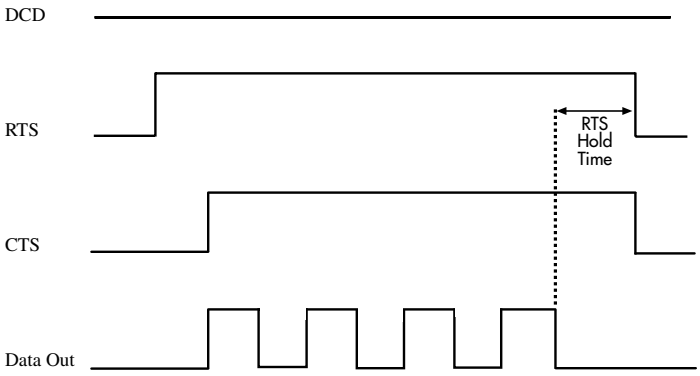


Figure 3: Transmit Data Logic Diagram

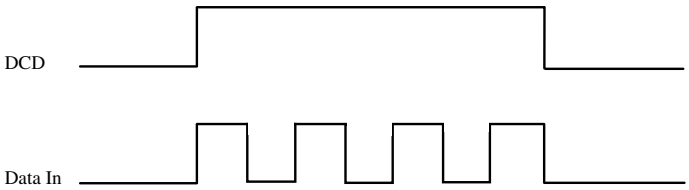


Figure 4: Receive Data Logic Diagram

The following definitions describe the timing parameters.

RTS Hold delay: RTS stands for request-to-send. This hardware signal tells the radio modem that we wish to transmit data. The RTS signal remains raised until all of the data has been sent. The RTS Hold delay is the amount of time that the RTS signal remains raised after the data has been sent. Typically this must be set to between 10-50ms, but to ensure reliable data transmission, you should check your radio modem specifications for this parameter.

CTS Timeout: CTS stands for clear-to-send. The radio modem will raise CTS when it is ready to transmit data. The CTS timeout is the amount of time the SCADA 3000 software will wait for CTS to be returned from the radio modem before it gives up.

DCD Time Out (Transmit): DCD stands for data-carrier-detect. The DCD signal indicates that the radio modem is receiving a carrier signal from another radio modem. The DCD transmit timeout is how long the software waits for DCD to drop before raising RTS when transmitting.

DCD Time Out (Receive): DCD receive timeout is how long the software waits for DCD to drop before it stops waiting for a packet to be received.

Flush Buffers after sending packet: Flush Buffers clears the send and receive buffers immediately after sending a packet. It can eliminate some noise but is not always necessary.

The following figure shows an illustration of how a typical radio network might look:

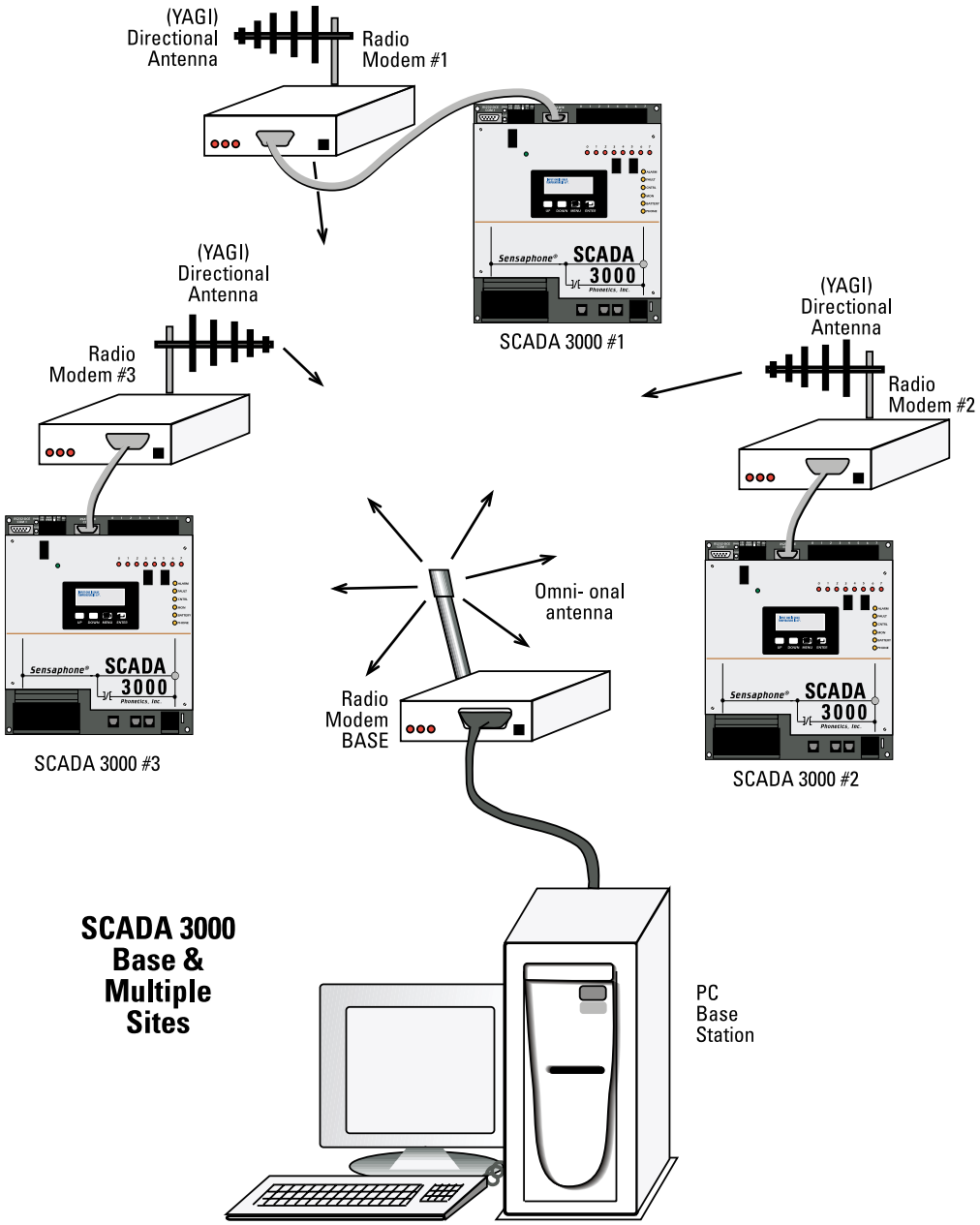


Figure 5: Radio network

COMMUNICATING WITH YOUR SCADA 3000s

The first step is to establish communications with each unit, one at a time, and monitor the communications statistics to ensure a low occurrence of errors. To monitor the communications, select *Comm*, then *Status* from the Sensaphone Control Center form. The following form will appear.

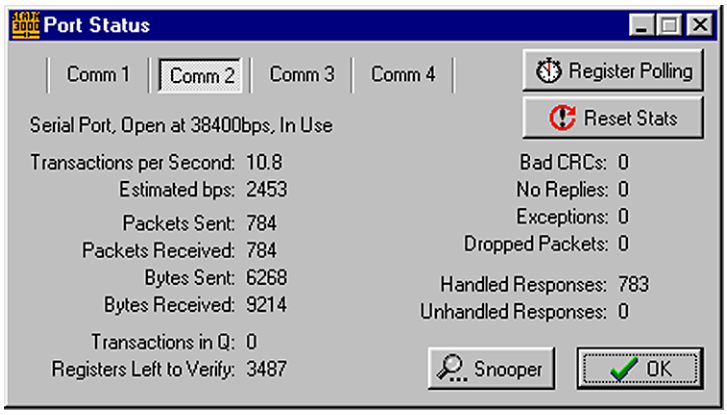


Figure 6: Port Status form

Good communication exists when the number of errors is kept to a minimum. To improve communications, try adjusting the modbus parameters or the Radio Modem settings from the Comm Setup form. Once reliable communications have been established, allow register verification to complete for each unit. This will ensure that the information displayed on the programming screens is accurate. At this point you can run a Real-time screen and monitor points from all of your units on one custom-designed screen. (See Chapter 14 for more information on real-time screens.)

RADIO MODEM MANUFACTURERS

Data-Linc Group

2635 151st Place N.E.
Redmond, WA 98052
Phone: (425) 882-2206
Fax: (425) 867-0865
Internet: www.data-linc.com

DataRadio COR Ltd.

Headquarters: Dataradio COR Ltd.
603 West Travelers Trail
Burnsville, MN 55337-2553 USA
Toll-Free: 1-800-992-7774
Headquarters Telephone: (612) 890-8155
Headquarters Fax: (612) 882-5671
Internet: www.johnsondata.com

FreeWave Technologies, Inc.

1880 South Flatiron Court
Boulder, CO 80301 USA
Phone: (303) 444-3862
Fax: (303) 786-9948
Internet: www.freewave.com

OMNEX Control Systems Inc.

#74 1833 Coast Meridian Road,
Port Coquitlam, B.C. V3C 6G5
Phone: 1-800-663-8806
Fax: (604) 944-9267
Internet: www.omnexcontrols.com

RF Neulink, A Wireless Solutions Company

7610 Miramar Road
San Diego, California 92126
Tel: (858) 549-6340 or 1-800-233-1728
Fax: (858) 549-6349
Internet: www.rfneulink.com

Scala Electronic Corporation

Post Office Box 4580
Medford OR 97501 (USA)
Phone: (541)779-6500
Internet: www.scala.net

Wi-Lan, Inc.

Head Office: 801 Manning Road NE
Suite 300
Calgary, Alberta T2E 8J5
Phone: (403) 273-9133
Toll Free: 1-800-258-6876
Fax: (403) 273-5100
Internet: www.wi-lan.com

Murandi Communications Ltd.

6715 8th Street NE
Suite 240
Calgary, Alberta, Canada T2E 7H7
Phone: (403) 777-9988
Fax: (403) 777-9989
Internet: www.murandi.com

Chapter 13: LCD & KEYPAD PROGRAMMING

The SCADA 3000 includes a 4-line by 20-character liquid crystal display (LCD) and a 4-button keypad on it's front panel. The display is used for viewing the values of data points, checking system information, and acknowledging alarms. It has two modes of operation: scrolling and manual. In scrolling mode the display will scroll through a programmable list of data points, showing 4 at a time. Every 5 seconds it will change to display the next set of points. You can customize the display to show only the points pertinent to your application. The keypad is used to manually view specific information, acknowledge alarms and program the data points that appear when the display is scrolling. You can also program the scrolling data points from the SCADA 3000 software.

Display Backlight

An LED backlight is integrated into the display to allow viewing in dark environments. To turn on the backlight, simply press one of the keypad buttons. The backlight will come on for a few minutes, then shut off automatically to save energy.

Programming Data Points for Scroll Mode

Click on the LCD button from the toolbar on the main programming form or select Programming then LCD from the main menu. The following form should appear:

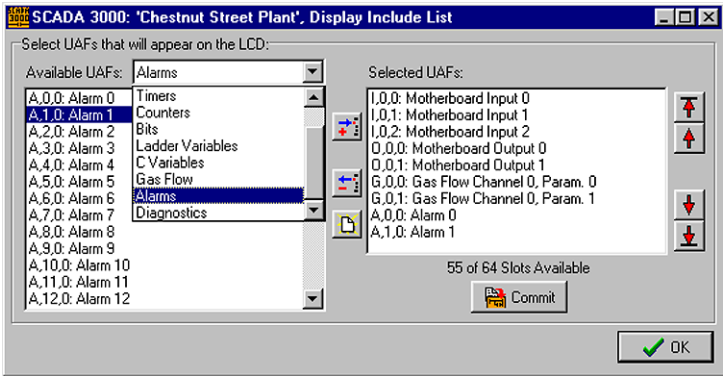


Figure 1: LCD Programming Form

This programming form is used to select the data points which will scroll on the LCD display. The box on the left provides a list of all the data points that can be chosen for display. The box on the right shows the data points that have been selected to appear on the display. In this process you will be choosing items from the list on the left and copying them to the list on the right. In the drop-down box next to the heading "Available UAFs", choose the data point category by clicking on the down arrow. The choices are: Inputs, Outputs, Timers, Counters, Bits, Ladder Variables, C Variables, Gas Flow, Alarms, and Diagnostics. When you pick a category a list of all possible points in that category will appear. Click on the item you wish to add to the scrolling display and then click on the Add to List button identified by the arrow pointing to the right between the boxes. This will copy the highlighted item to the list of Selected Data Points on the right (e.g. those to appear on the LCD). Continue this process by highlighting each data point on the left and copying it to the list of Selected Data Points on the right. When you have finished making your list, click on the Commit button. This will upload the list to the SCADA 3000 and it will immediately start displaying the programmed data points

on the display. Note that the unit defaults to having all of the main unit inputs and outputs selected for display.

Removing Data Points

To remove data points from the LCD, click on the point you wish to remove from the list of Selected Data Points from the box on the right. Next, click on the Remove from List button identified by the arrow pointing to the left, located between the boxes. To remove all of the Selected Data Points from the list, click on the Remove All button, located at the bottom between the two boxes.

Grouping Data Points

The selected data points will be displayed four at a time. This means that the first four points in the list will be displayed together, followed by the next four points, etc... The selected data points can be rearranged so that they are grouped in an orderly fashion. The up/down arrows on the far right of the form provide the ability to move data points up and down in the list. To rearrange the Selected Data Points, highlight the data point you would like to move by clicking on it, or drag the mouse over a group of data points, then click on one of the arrows on the right. The interior arrows will move the data points up or down one place and the outside arrows will move them to the extreme top or bottom of the list. When finished editing the list, click on the Commit button.

NAVIGATING THE DISPLAY USING THE KEYPAD

The SCADA 3000 window displays the various menu selections you choose with the buttons below it. The main menu choices--displayed by pressing the MENU button--are DATA POINTS, ALARMS, SYSTEM STATUS, and RETURN. The figure below shows a typical display for Input Status.

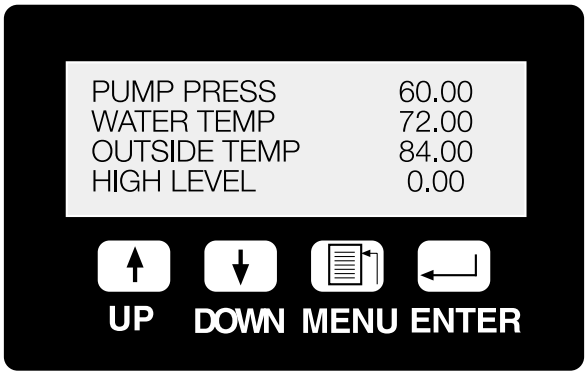


Figure 2: SCADA Display Window

Scrolling Display

The SCADA 3000 will normally scroll through a programmed list of I/O points and variables that you select. It defaults to the 16 inputs and 8 outputs on the main board. You can add or remove items to the scrolling list via the keypad or through the Programming Software. From the keypad you would use the *Add to Display* and *Remove from Display* functions.

If you want to go back a menu or just want to return to the programmed scrolling values, keep pressing the MENU button. Each time it is pressed, the display will go back a menu until you

reach the main menu. The last item in the main menu is RETURN. Selecting this option will return the display to its normal scrolling routine.

Use the Up/Down arrows to select between ADD, REMOVE, and DISPLAY FROZEN. To Add or Remove, press the ENTER key while the corresponding message is shown. To freeze the display, simply press the arrows until the message DISPLAY FROZEN is shown. Do not press ENTER. The display will stay on this screen as long as the message DISPLAY FROZEN is shown. To return to the main menu, press the MENU key several times until it is shown.

USING THE KEYPAD

The keypad consists of four buttons labeled from left to right "UP," "DOWN," "MENU," and "ENTER." To navigate the SCADA program, you press MENU for the main choices listed above, and then use the ENTER and UP and DOWN keys to navigate through each program option in the main menu and all embedded menus. When the cursor > is at the item you want, press ENTER. The options operate as follows.

DATA POINTS

Press MENU to see the list. The cursor > will appear automatically to the left of the first item in the list, DATA POINTS. Press ENTER to view the options for Data Points. You will then see the following list of options: INPUTS, OUTPUTS, TIMERS, COUNTERS, BITS, LADDER VARIABLES, C PROGRAM VARIABLE, GAS FLOW, and DIAGNOSTICS.

Inputs

Press ENTER for the first selection under Data Points, Inputs. Then use the arrow keys to navigate up and down the list of inputs, to see those below the number displayed on the screen. These are numbered as MODULE ADDRESS 0 through MODULE ADDRESS 15. Scroll down or up and press ENTER for any of these to display the INPUT CHANNEL list. Scroll through this list and press ENTER for any Channel to view its current value or to execute the Add, Remove or Freeze functions. For each, using the arrow keys, you can scroll through 3 menu choices: ADD TO DISPLAY, REMOVE FROM DISPLAY, or DISPLAY FROZEN.

The names for the channels are customizable with your software.

Outputs

Use the DOWN arrow and ENTER to select the Outputs display. The list goes from MODULE ADDRESS 0 to MODULE ADRESS 15. Scroll down or up and press ENTER for any of these to display the OUTPUT CHANNEL list, 0-7. Scroll through this list and press ENTER for any Channel to view its current value or to execute the Add, Remove or Freeze functions. For each, using the arrow keys, you can scroll through 3 menu choices: ADD TO DISPLAY, REMOVE FROM DISPLAY, or DISPLAY FROZEN. The Outputs are likewise customizable to display the identifier of your choice.

Timers

The third option under Data Points is Timers. The choices are TIMER NUMBER 0-63. Use the down arrow to scroll through the timers below those displayed. Press ENTER for any of the timers to bring up the TIMER PARAMETER feature. There are five parameter choices, selected by using the arrow and ENTER keys. Press ENTER for any of parameter to view the current value or to execute the Add, Remove or Freeze functions. For each address, using the arrow keys, you can scroll through 3 menu choices: ADD TO DISPLAY, REMOVE FROM DISPLAY, or DISPLAY FROZEN.

Counters

The fourth option under Data Points is Counters. The choices are COUNTER NUMBER 0-63. Use the down arrow to scroll through the counters. Pressing ENTER on any will bring up a COUNTER PARAMETER list. Select any of the seven parameters by pressing ENTER; this will display the current value or to execute the Add, Remove or Freeze functions. For each, using the arrow keys, you can scroll through 3 menu choices: ADD TO DISPLAY, REMOVE FROM DISPLAY, or DISPLAY FROZEN.

Bits

The fifth option is Bits. The display offers a scrolling WORD NUMBER list from 0-255. Select any of these by pressing ENTER to view the scrolling BIT NUMBER list from 0-15. Again, pressing ENTER for any Bit Number will reveal the current value or to execute the Add, Remove or Freeze functions. For each, using the arrow keys, you can scroll through 3 menu choices: ADD TO DISPLAY, REMOVE FROM DISPLAY, or DISPLAY FROZEN.

Ladder Variables

The next option is Ladder Variables. The display offers a scrolling LADDER VARIABLE list from 0-127. Press ENTER on any Number to view the current value or to execute the Add, Remove or Freeze functions. For each address, using the arrow keys, you can scroll through 3 menu choices: ADD TO DISPLAY, REMOVE FROM DISPLAY, or DISPLAY FROZEN.

C Program Variable

Next is the C-Program Variables list. Again, the display offers C VARIABLE from 0 to 1023. Press ENTER on any number to view the current value or to execute the Add, Remove or Freeze functions. For each address, using the arrow keys, you can scroll through 3 menu choices: ADD TO DISPLAY, REMOVE FROM DISPLAY, or DISPLAY FROZEN. Hit ENTER to select.

Gas Flow

The Gas Flow menu allows you to select from a list of 20 gas flow parameters per Gas Flow Channel. First select the Gas Flow Channel from the list of 0-3. Next, select the Gas Flow parameter from the list of 0-19. Press ENTER for any parmeter to view the current value or to execute the Add, Remove or Freeze functions. Use the Up/Down arrows to select between Add, Remove or Freeze.

Diagnostics

The diagnostics option offers a list of internal diagnostic information that can be added to the scrolling display. Select Diagnostics and you are given a list of Modules. The first is Module 0, the main unit. Modules 1-15 are for additional SCADA modules you have attached to the main unit. The diagnostics list for the Main Unit is much longer than the list for the I/O modules. For the main unit you scroll through the following list.

Month
Day
Year
Hours
Minutes
Seconds
Input Ref

Voice Ref
3V Battery
Battery
Main Power
15V Supply
24V Supply
Unit Temp
Current Usage
Com Rom
Cntl Rom
Board Rev
Comm Status
Cntl Status
Watchdog
Comm Reset
Cntl Reset

For individual modules 1-15, the list of diagnostics reads:

Module Type
revision
Month
Day
Year
Stack
Timeouts

There are four additional spaces at the end for future modifications to diagnostic programming for the I/O modules.

Comm Port Baud Rate

The baud rate of each of the unit's serial ports can be displayed from the keypad as well. To display the baud rate, press MENU, then select Data Points and press ENTER. Press the down arrow key until you reach Diagnostics, then press ENTER. Select Module 0 and press ENTER, then press the down arrow key until you reach Com1 and Com2 baud rate. Press ENTER.

Diagnosing Faults

If the fault LED is ON you can find out why by pressing MENU, then selecting DATA POINTS by using the arrow keys. Press ENTER. Press the down arrow key until you reach the last entry, SYSTEM FAULTS, and press ENTER. Any system faults will be displayed.

ALARMS

The second selectable item on the main menu is Alarms. Use the down arrow to move the cursor to the left of Alarms, and press the ENTER keypad button.

A list appears of the 0-63 possible alarm points and the current alarm status for each. You can scroll through the alarm list with the up and down arrow keys and press ENTER to view any specific alarm point's current value or to execute the Add, Remove or Freeze functions. The first three lines of the display are: The alarm Name, the alarm UAF, and the alarm Status. An example appears below:

```
Alarm 46
A, 46, 0
No alarm exists
```

The fourth line cycles through five choices as you press the arrow keys. The choices are: Add to Display, Remove from Display, Acknowledge All, Acknowledge Alarm, Display Frozen.

Acknowledging Alarms from the Keypad

One or more alarms can be acknowledged at the keypad to keep the unit from dialing when personnel are on site. The procedure below steps through the key sequence.

1) Press the MENU button, arrow-down to the Alarms category, and press ENTER.

2) Now press the down-arrow key until you reach the alarm you want to acknowledge. Press ENTER.

3) Press the down-arrow key until the LCD displays *Acknowledge Alarm*. Press ENTER and the alarm will be acknowledged. If you continue to press the down-arrow key, the *Acknowledge All* option will be displayed. If you press ENTER while this message is showing, all alarms will be acknowledged.

SYSTEM STATUS

The third choice on the main menu, System Status presents a four lines of data: Date/Time, Programs Running, Datalog activity, and Alarm status. More comprehensive system status information is available through the software program.

RETURN

The fourth menu choice is Return. Return presents the "normal" view display--a scrolling list of the main unit's I/O points and, as the name implies, the information being returned by them.

Chapter 14: REAL-TIME SCREEN DESIGN

SUMMARY

The Real-time Screen feature allows you to design a graphical representation of your process and have your I/O values display on the screen in real time. Several sample screens are included to give you some ideas and to demonstrate the various features of the software. Building your own real-time screen requires two steps:

1. Creating the screen using the real-time screen designer

2. Running the screen that actually logs into the selected units on the screen and begins updating the I/O points in real time.

EDITING

To run the Real-time Screen Designer, click *Real-time* on the Control Center menu bar. From the pulldown menu, select Editor. This will bring up the Real-time Screen design menu, the RT Builder. From its menu bar, select *File*, and then *New* to start designing a new screen.

SCADA 3000 RT Builder
File Edit Forms Help
System DataPoint Graphic Group Contain Rotate Elec Gauge Valve AudVid Dynamic
New_1

Figure 1: RT Builder form

> **Note**: The R.T. button on the toolbar will open Real-time in Run mode only, which requires you to have a real-time program already designed.

In general, you will select items from the main RT Builder tabs and click-drop them onto the palette, then program properties for each one.

> **Note**: There are two ways to place an object on the palette. You can simply double click on it where it appears in the tabs and it will drop onto a default position on the palette, or you can click once on an icon to select it and then click once on the palette where you want it to appear.

The RT Builder toolbar contains 11 tabs: Data, Grouping, Graphics, DataPoint, Electrical, Rotating, Container, Valves, Gauges, AudVid, DynComponents.

Once you choose *New*, two forms will appear on your screen. The first of these is your Palette. This is the platform on which you actually paste in and assemble the graphical elements of your real-time screen. In editing mode, this is defined by a grid, which disappears in run mode.

Sizing the Palette

You can adjust the size of the palette in design or run mode by dragging the edge of the form to the desired height, width and position; alternatively, you can specify the size of the palette in the palette properties menu. To maximize the palette screen, double-click on the bar at the top of the palette.

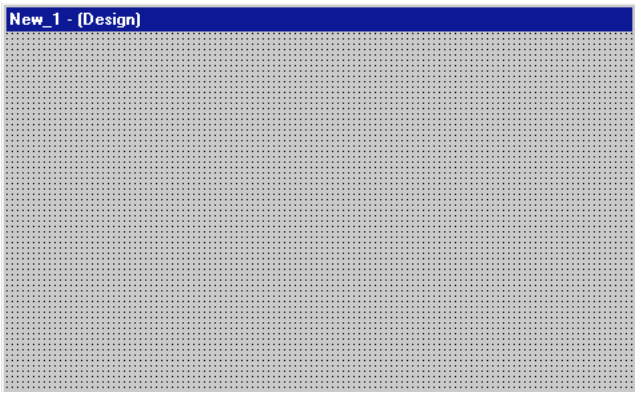


Figure 2: Palette

The second form is the Property Editor.

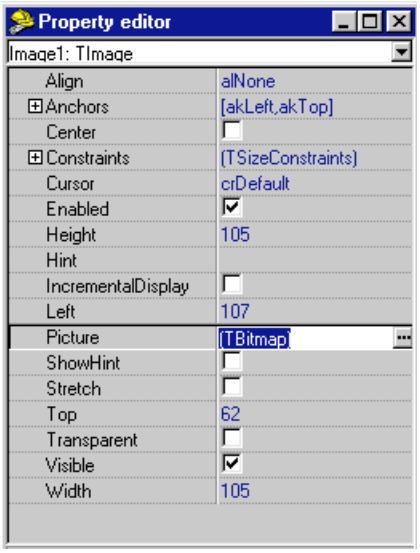


Figure 3: Property Editor

Each element of the design process has properties that can be customized to suit your application. The properties will differ depending on the element selected—for example, the palette, a container, a bitmap, a dynamic component. Beside some properties listed in the properties editor, you will see a white box with a + sign. This box indicates the existence of sub-properties associated with the particular property. Click on the white box to expose the sub-properties list.

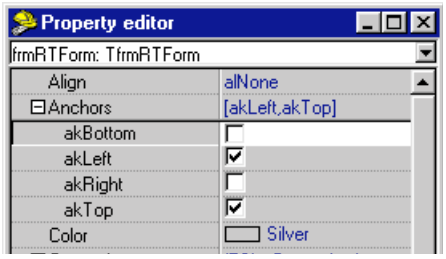


Figure 4: Sub-properties

Two levels of properties can be displayed for all elements: basic and advanced. The advanced properties provide additional capabilities that some users may require. To display advanced properties, go to the *File* menu and click on *Advanced Properties*.

A description of how each of the properties affects each element of the realtime screen is provided throughout the sections following.

Palette Properties

To edit the palette properties, select the palette as the object to be edited by clicking on an empty area of the grid. In the first box of the property editor form, the word *frmRTForm: TfrmRTForm* will appear. The properties listed on the form will only affect the palette. A description of the basic and advanced palette properties is listed below.

Basic Properties

Color: Sets the color of the palette.

Height: Sets the total height of the palette

Left: Sets the screen position of the palette as measured from the left side of the monitor.

Title: Text that appears at the top of the palette.

Top: Sets the screen position of the palette as measured from the top side of the monitor.

Width: Sets the total width of the palette.

Advanced Properties

Font: Sets the default font for labels and text. Click in the font box and then click the small button at the end of the box to select a font.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the Size property.

SELECTING SCADA UNITS

You may retrieve and display I/O information from one or more SCADA 3000 units on a single Real-time screen. To do this, you must define which units will be opened when the screen is run. To select SCADA units click on the *Data* tab. Next, click on the SCADA 3000 icon and then click on the palette. The SCADA icon will appear on the palette. (It becomes invisible in Run mode.) Use the Property Editor to assign the icon to a unit in your database. Click on SCADA Unit in the Property Editor to assign the icon to a unit in your database. Click on *SCADAUnit* in the Property Editor and click on the button at the end of the box. This will bring up a form to select a SCADA 3000 from your database.

Use the pulldown menu to the right to bring up the list of possible units. Select the unit you're configuring. Once you select a unit, the Active box should show a checkmark.

Click on the *ConnectType* box above SCADAUnit in the Property editor and select the type of connection for the unit. (See figure below.) You have three choices:

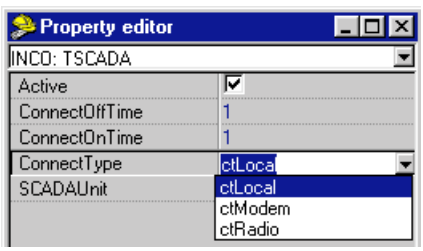


Figure 5: Connect type menu

Local: Means that the unit is connected directly to your computer via a serial port.

Modem: Indicates that your computer must dial into the unit via your modem. When more than one unit requiring a modem connection is selected on the real-time screen, the software will periodically connect to each unit. Use the ConnectOnTime and ConnectOffTime slots to specify how long your modem should connect to the selected unit(s). ConnectOffTime is the intercall time from when the selected unit disconnects until the next unit connects. The parameters of each are specified in minutes.

> **Note**: If you are connecting to only one unit via modem and want the connection to be continuous, select Local as the connect type.

Radio: Means that you are communicating with your units via radio modem. In this scenario, the computer will poll the selected I/O points from each unit on a continuous basis.

When the Real-time screen runs, it will connect to all units connected via serial port first, followed by units on a radio network, and lastly units connected via modem. The order of calls for modem units is based on the order in which they were added to the realtime screen during the initial design stage.

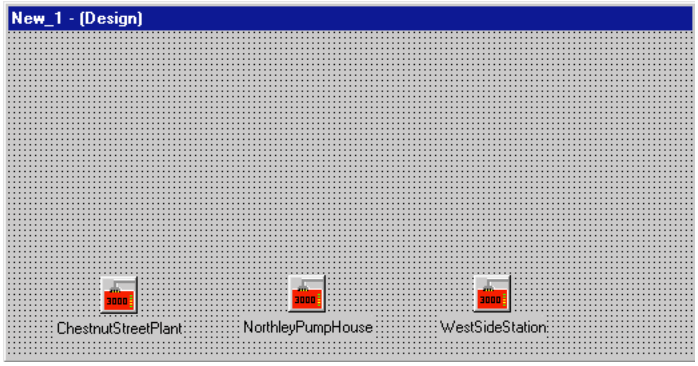


Figure 6: Palette with connected units

PLACING OBJECTS ON THE PALETTE

A variety of standard objects can be accessed with the tabbed categories: Electrical, Rotating, Containers, Valves, and Gauges. To place objects on the Palette, select one of the tabs and click on an object. For example, from the Container Tab, click on the *ContainerVessel* object. Now click on the palette. The Vessel should appear on the palette as shown below.

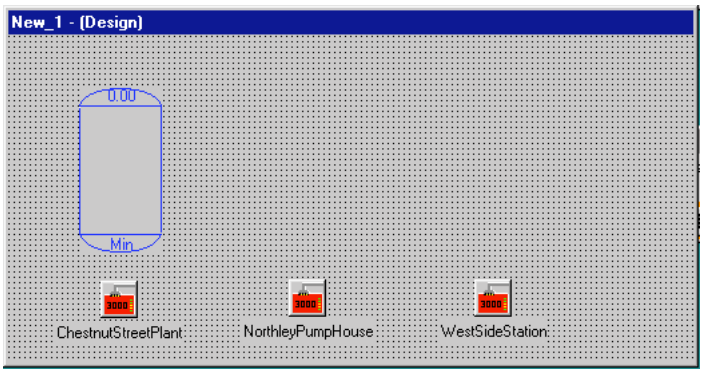


Figure 7: Vessel object

Next, edit the properties of the vessel to assign it to an I/O value. Adjust the minimum and

maximum values, set the colors, and so forth. A list of properties with descriptions is provided below.

Basic Properties

BackGndMode: There are 2 settings: *bmOpaque* and *bmTransparent*. The *bmOpaque* setting will display the background color of the component or image as specified by the Color property. The *bmTransparent* setting will display the component background as transparent so that any component, image or background will be visible.

Color: Background color of component when set to *bmOpaque*.

Datapoint: Assigns the component to an I/O point within a SCADA unit. Reveal the sub-properties by clicking the white box next to the word *Datapoint*. Select a SCADA unit first, then select the datapoint.

Datapoint: Enter the datapoint UAF or click in the Datapoint box above and then click the button at the end of the box to display the Datapoint selection form.

DisplayPercent: Displays the datapoint value as a percent of the MinValue and MaxValue (see these below).

DisplayValue: Displays the actual datapoint value.

FillColor: Selects the fill color for components which display the level as a moving bar.

FillStyle: Provides options to have the fill be solid or to have various patterns.

Font: Sets the font for the selected object

Charset: Specifies the character set for the font. Dependent upon what values of Charset are valid as supplied by the font vendor.

Pitch: Pitch can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used for the font. The *fpFixed* and *fpVariable* values can be used to set the pitch explicitly. A value of *fpFixed* causes each character in the font to have the same width, while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object.

Style: Sets the default style of the font or text.

fsBold: Sets the font to be boldfaced.

fsItalic: Sets the font to italic.

fsStrikeout: Forces a line through the characters.

fsUnderline: Underlines the text in that font set.

Height: Sets the height of the object.

Left: Sets the position of the object as measured from the left side of the palette.

MaxValue: Sets the maximum value of the fill.

MinValue: Sets the minimum value of the fill.

StayOnTop: Enables the object to stay on top when it overlaps other objects on the screen.

Top: Sets the position of the object as measured from the top of the palette.

Value: Sets the datapoint value in design mode to view its effect on the object. (e.g., fill level on/off state)

Width: Sets the width of the object.

Advanced Properties

Align: Stretches the object to align with the top, bottom, left, right, or client part of the palette.

Anchors: Sets one or more sides of the object as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the object a fixed distance from the bottom of the palette.

akLeft: Anchors the object a fixed distance from the left side of the palette.

akRight: Anchors the object a fixed distance from the right side of the palette.

akTop: Anchors the object a fixed distance from the top of the palette.

Constraints: Limits the maximum and minimum size of the object.

MaxHeight: Sets the maximum height of the object. (0 disables this property)

MaxWidth: Sets the maximum width of the object. (0 disables this property)

MinHeight: Sets the minimum height of the object. (0 disables this property)

MinWidth: Sets the minimum width of the object. (0 disables this property)

Hint: A text message which appears when the cursor rests over an object.

Show Hint: Enables the hint message to display.

Visible: Makes the object visible or invisible.

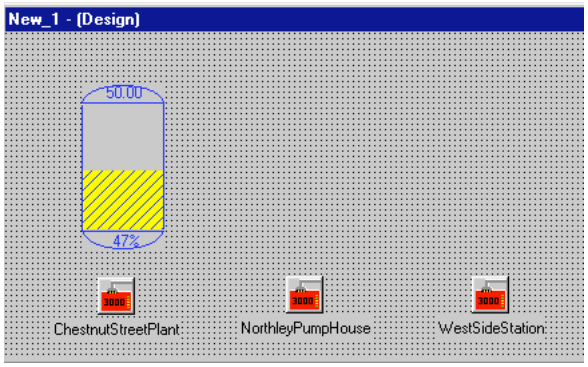


Figure 8: Vessel object displaying set I/O properties

Right-click Tools

By right-clicking your mouse anywhere on the palette, you can bring up a menu containing short cuts for customizing your palette and the objects on it.

Figure 9: Right-click Menu

Builder Form: Switches the foreground window to the RT Builder.

Properties:

Edit Image (graphics): Can be used as a shortcut to grabbing BMP files.

Delete (objects): Removes the object from the palette.

Cut (objects): Cuts the object to the clipboard for pasting elsewhere.

Copy (objects): Makes a copy of the object to paste elsewhere.

Align to grid: Aligns the top left corner of selected object to the underlying palette grid.

Bring to Front: Moves the selected object to the foremost position.

Send to Back: Moves the selected object behind any others that overlay it.

Align: Brings up the following Menu.

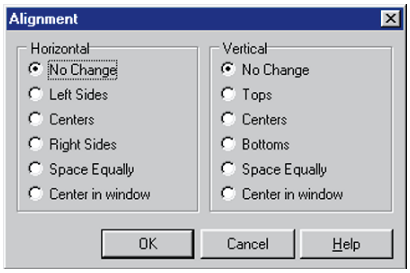


Figure 10: Alignment Menu

Align multiple objects horizontally or vertically and relative to each other by clicking in the appropriate alignment option boxes.

Size: Brings up the following menu.

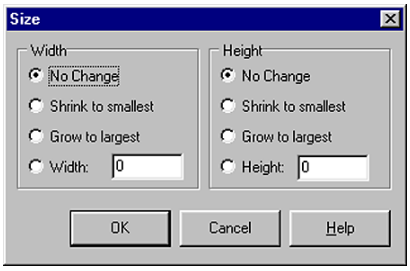


Figure 11: Size Menu

Use the radio buttons to select Width or Height sizing variations, or type in the specific width or height you desire. The number you type in represents the number of pixels in width or height.

Scale: Allows you to scale the size of the selected object by percentage. Type in the percentage and click OK.

Tab Order: Use the up and down menus to move objects higher or lower in the list.

Creation Order: Use the up and down menus to move objects higher or lower in the list.

GROUPING COMPONENTS IN A BOX, PANEL OR PAGE

The Group Tab offers three choices for framing/grouping components on your palette.

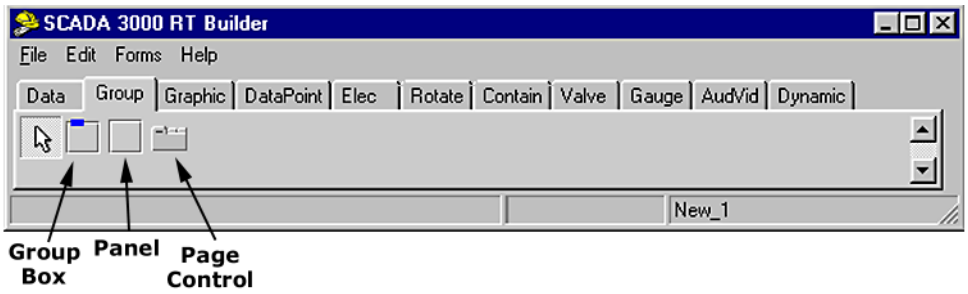


Figure 12: Group Tab

GroupBox creates a frame within which you can place any graphic components from the other tabs. These will remain within the box and will move when the box itself is dragged around the screen. The group box has a text identifier with which you can name it.

Panel is similar to a group box but without the identifier. It's a frame into which you can fit various graphic components. The appearance of the panel can be changed from the Propery Editor, including a variety of bevels.

Page Control is most useful if you have multiple sites or devices/conditions that exceed what comfortably fits on one screen. If you Right-Click on a Page Control object and choose *New Page* from the popup menu, you can create and add multiple tabs, which can be resized to create multiple full-palette screens. Each one can have its own graphical components and display properties. For example, if you monitor multiple pumping stations, you can assign the components of each a separate tabbed page, name each page to identify it, and even include a summary page for a graph comparing them.

Grouping Properties

Active Page (PageControl only): Select the active page.

Align: To align panel/page to edges of the palette area.

alBottom: Stretches the panel across the bottom of the palette.

alClient: Fills the palette with the panel.

alLeft: Stretches the panel along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the panel along the right edge of the palette.

alTop: Stretches the panel across the top of the palette.

Alignment (Panel only): Sets the justification of the text caption.

Anchors: Sets one or more sides of the box as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the box a fixed distance from the bottom of the palette.

akLeft: Anchors the box a fixed distance from the left side of the palette.

akRight: Anchors the box a fixed distance from the right side of the palette.

akTop: Anchors the box a fixed distance from the top of the palette.

AutoSize (Panel only): Check to fit the containing box to the elements it contains.

Bevel Inner (Panel only): Adjusts the appearance of the inner bevel.

Bevel Outer (Panel only): Adjusts the appearance of the outer bevel.

Caption: Name for the box, panel or page.

Color: Color of the group container. (Panel only: A custom list of colors for button surfaces.)

Constraints: Limits the maximum and minimum size of the box.

MaxHeight: Sets the maximum height of the box. (0 disables this property)

MaxWidth: Sets the maximum width of the box. (0 disables this property)

MinHeight:Sets the minimum height of the box. (0 disables this property)

MinWidth: Sets the minimum width of the box. (0 disables this property)

Font: Sets the default font (TFont). Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font. Dependent upon what values of Charset are valid as supplied by the font vendor.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used with the font. *fpFixed* causes each character in the font to have the same width (monospacing). *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Vertical size in pixels (over-ridden if you chose a Max/Min Height).

Hint: Text shown on mouseover.

Left: X coordinate of left edge of component on screen.

Multiline (PageControl only): Allows multiple levels of tabbed pages.

Ragged Right (PageControl only): Sizes tab to text width when tabwidth is set to 0.

Show Hint: Check to activate hint feature.

Style (PageControl only): Allows tabs to appear as buttons, flat buttons or tabs.

TabHeight (PageControl only): Adjusts the tab height. Automatically adjusts to text when set to 0.

TabWidth (PageControl only): Sets the tab width. Automatically adjusts when set to 0.

Top: Y coordinate from top edge of component on screen.

Visible: Check to make box visible.

Width: Horizontal size of box/panel in pixels (over-ridden by Max/Min Width values, if any.)

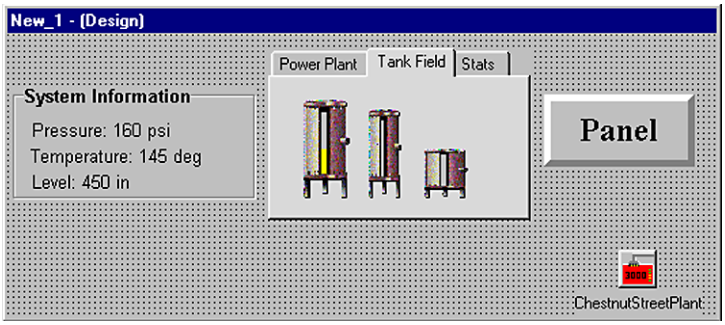


Figure 13: Examples of Box, Page & Panel

PLACING A GRAPHIC ELEMENT ON THE PALETTE

The Graphic Elements are: Label, Image, Shape and Bevel.

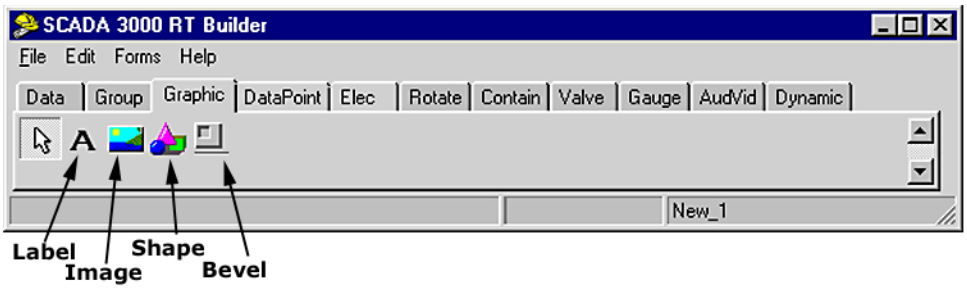


Figure 14: Graphic Tab

Label creates a static label you can use to name any object on your screen or place any other unchanging text on your palette. You can alter size, font and color. Labels can be placed either adjacent to or on top of images. **Image**: You can select custom bitmap representations of many more shapes, containers and devices than are listed on the tabs, including vector-drawn images. Clicking on *Image* places an empty square on the palette surface. From the Property Editor, select *Picture*. Use the pulldown menu to open an Image form. Select *Open* (the open folder button) and locate your real-time bitmap collection. Choose an image you want to place on the palette. See the figure below.

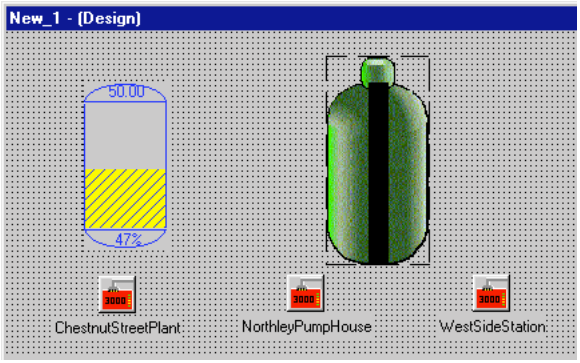


Figure 15: Bitmap tank graphic placed on palette

Note: You can add your own graphics to those supplied. The images should be saved as 8-bit uncompressed BMPs.

Select *Stretch* from the Property Editor to make the graphic fit into the space on your palette. You can pull, distort and shrink the bitmap to make it fit as you like. Choose *Transparent* if you want to make the plain white background invisible.

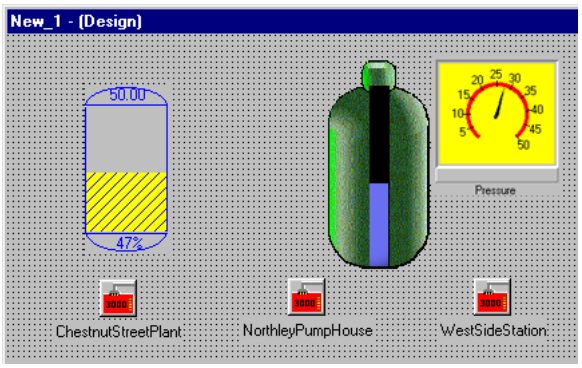


Figure 16: Bitmap graphic linked to I/O, Progress Bar and AGauge

You can make bitmap images more animated by adding Dynamic Labels and GaugeLevel objects. Dynamic labels are numerical labels that represent I/O values and update continuously. GaugeLevel objects are dynamic bar gauges that can be placed on top of images to indicate high/low levels. The example below shows a bitmap image of a well with a dynamic label showing the level in feet, a patterned ContainerFill, and static labels.

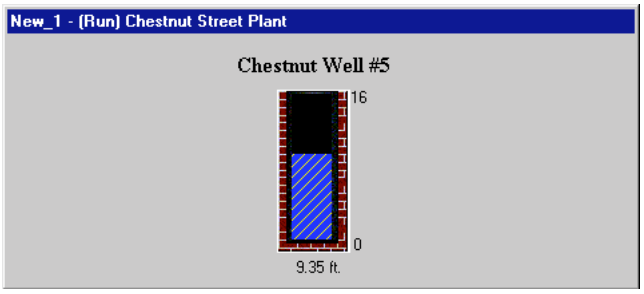


Figure 17: Bitmap Well with GaugeLevel

The third button on the Graphics tab, **Shape**, places a simple geometrical shape on your palette. On the Propery Editor, select the Shape box to change to a different shape. The default shape is a rectangle.

Bevel, the last button, creates a beveled shape. You use *Shape* and *Style* on the Property Editor to alter the beveling.

Properties

Align: To align graphic/component to edges of the palette area.

alBottom: Stretches the component across the bottom of the palette.

alClient: Fills the palette with the component. Useful for adding a background graphic.

alLeft: Stretches the component along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Alignment: *taCenter*, *taLeftJustify*, *taRightJustify*. Aligns the text in the label/box.

Anchors: Sets one or more sides of the graphic/label as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the element a fixed distance from the bottom of the palette.

akLeft: Anchors the element a fixed distance from the left side of the palette.

akRight: Anchors the element a fixed distance from the right side of the palette.

akTop: Anchors the element a fixed distance from the top of the palette.

Autosize (Label): Automatically sizes the background box to the size of the text.

Brush(Shape):

Color: Color fill for the shape.

Style: Fill patterns for the shape.

Caption (Label): Name or text assigned to graphic.

Center (Image): Checkbox.

Color: Background color (of Label or element if applicable)

Constraint: Limits the maximum and minimum size of the label/graphic.

MaxHeight: Sets the maximum height. (0 disables this property)

MaxWidth: Sets the maximum width. (0 disables this property)

MinHeight:Sets the minimum height. (0 disables this property)

MinWidth: Sets the minimum width. (0 disables this property)

Font: Sets the default font for labels and text. Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: Can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used with the font. The *fpFixed* and *fpVariable* values can be used to explicitly set the pitch; a value of *fpFixed* causes each character in the font to have the same width (monospacing), while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Vertical size in pixels (over-ridden if you chose a Max/Min Height).

Hint: Text shown on mouseover.

Incremental Display (Image): Checkbox.

Layout (Label): *tlBottom*, *tlCenter*, *tlTop*. Locates the label text to the top, bottom or center of the box.

Left: X coordinate of left edge of graphic or label on screen.

Pen(Shape):

Color: Selects the border color of the shape.

Mode: *pmBlack*, *pmCopy*, *pmMask*, *pmMaskNotPen*, *pmMaskPenNot*, *Merge*,

pmMergeNotPen, *pmMergePenNot*, *pmNop*, *pmNot*, *pmNotCopy*, *pmNotMask*, *pmNotMerge*, *pmNotXor*, *pmWhite*, *pmXor*.

Style: Solid or Dashed lines.

Width: Selects the width of the shape border.

Picture(Image): Select a bitmap graphic to appear on the palette.

Shape(Bevel): bsBottomLine, bsBox, bsFrame, bsLeftLine, bsRightLine, bsSpacer, bsTopLine.

Shape(Shape): *stCircle*, *stEllipse*, *stRectangle*, *stRoundRectangle*, *stRoundSquare*, *stSquare*.

Show Hint: Check to show text on mouseover.

Stretch(Image): Check to be able to resize graphic.

Style(Bevel):

bsLowered: a lowered bevel.

bsRaised: a raised bevel.

Top: Y coordinate from top edge of graphic element on screen.

Transparent: Check to make the background of a bitmap image invisible.

Visible: Check to make the element visible.

Width: Horizontal size in pixels (over-ridden by Max/Min Width values, if any.)

Word Wrap (label): Check to make text wrap around to the next line within a label text box.

PLACING DATAPOINT ELEMENTS ON THE PALETTE

The DataPoint Tab offers four element buttons you can use to place I/O related elements on the palette.

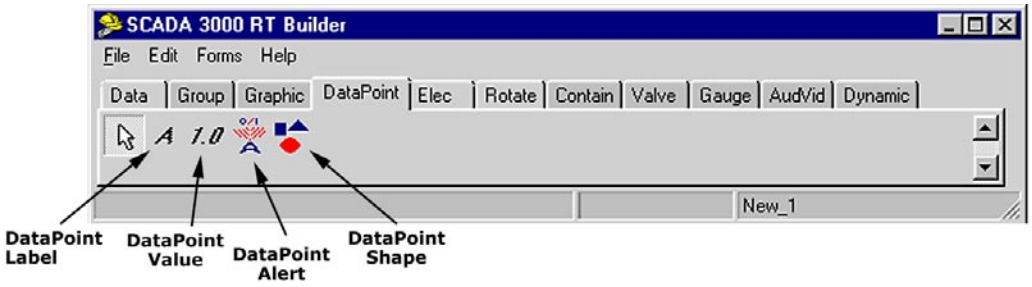


Figure 18: DataPoint Tab

The **DatapointLabel** is a dynamic label which can be assigned to the I/O name of any point in the SCADA 3000. For example, if the label were assigned to input 0 on the main unit (UAF I,0,0), whatever name was entered for Input 0 in the SCADA 3000 would now appear on the screen.

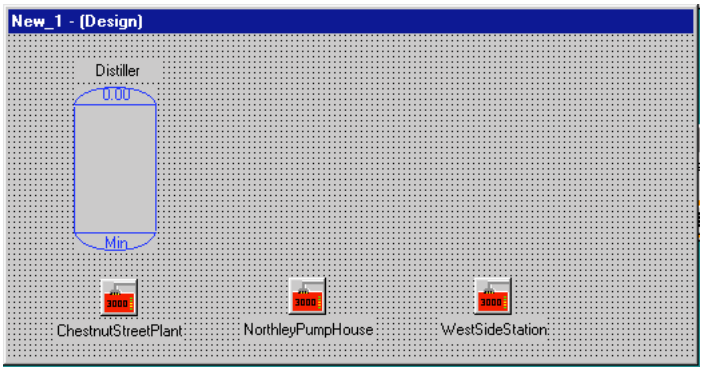


Figure 19: Label attached to vessel object

Labels can be used to customize the screen. You can edit the size, font and color of the text. You can place DataPointValue labels directly on top of graphics to provide real-time readings. Label properties are listed below.

Properties (Label/Value)

Align: To align graphic/component to edges of the palette area.

alBottom: Stretches the component across the bottom of the palette.

alClient: Fills the palette with the component. Useful for adding a background graphic.

alLeft: Stretches the component along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Alignment: Aligns the text in the box: center, left justified, right justified.

Anchors: Sets one or more sides of the graphic/label as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the element a fixed distance from the bottom of the palette.

akLeft: Anchors the element a fixed distance from the left side of the palette.

akRight: Anchors the element a fixed distance from the right side of the palette.

akTop: Anchors the element a fixed distance from the top of the palette.

Autosize: Automatically sizes the background box to the size of the text.

Caption (Value): Name or text assigned to graphic.

Color: Sets the background color of the text box.

Constraints: Limits the maximum and minimum size of the label/graphic.

MaxHeight: Sets the maximum height. (0 disables this property)

MaxWidth: Sets the maximum width. (0 disables this property)

MinHeight:Sets the minimum height. (0 disables this property)

MinWidth: Sets the minimum width. (0 disables this property)

Datapoint: Assigns the component to an I/O point within a SCADA unit. Reveal the sub-properties by clicking the white box next to the word *Datapoint*. Select a SCADA unit first, then select the datapoint.

Datapoint: Enter the datapoint UAF or click in the *Datapoint* box above and then click the button at the end of the box to display the Datapoint selection form.

SCADA: Select the RTU being selected.

Font: Sets the default font for labels and text. Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font. Dependent upon what values of Charset are valid as supplied by the font vendor.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: Can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used with the font. The *fpFixed* and *fpVariable* values can be used to explicitly set the pitch; a value of *fpFixed* causes each character in the font to have the same width (monospacing), while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Sets the height of the text box.

Hint: Text shown on mouseover.

Layout: Sets the position of the text within the box: centered, top or bottom justified.

Left: Sets the position of the text box as measured from the left side of the palette.

Show Hint: Check if you want the mouseover hint to appear.

Top: Sets the position of the text box as measured from the top of the palette.

Transparent: Makes the background color of the text box transparent.

Visible: Check to make the element visible.

Width: Sets the width of the text box.

Wordwrap: Makes text within the text box wrap around to the next line if it exceeds the box width.

The **DataPointValue** displays the current numerical value or state of the specified I/O. (See Properties listed above)

The **DataPointAlert** can be used to change or set a property of another element on the palette when an I/O value changes from On to Off or crosses a threshold. As an example, you could place the DataPointAlert and the WavPlayer elements on the palette. Assign the DataPointAlert to an I/O value and set the threshold type and threshold value properties to determine when to play a Wav file. The *DynComponent* property selects the element on the palette you wish to change, in this case the WavPlayer. The *DynProperty* selects the property of the DynComponent you wish to change. In our example you would select *Play*, because we want the Play property to

be set when the DataPointAlert is tripped. You can use the DataPointAlert in many ways to affect changes in the realtime screen components. (Note that depending on the property changed, the property may not revert back to its original value after the DataPointAlert returns to its Off value.)

Properties

AlertInterval: Determines how often (in seconds) to set the target property.

Datapoint: Assigns the component to an I/O point within a SCADA unit. Reveal the sub-properties by clicking the white box next to the word *Datapoint*. Select a SCADA unit first, then select the datapoint.

Datapoint: Enter the datapoint UAF or click in the *Datapoint* box above and then click the button at the end of the box to display the Datapoint selection form.

DynComponent: Selects the object or element on the Palette to be modified.

DynProperty: From the component selected above, this setting selects the property to be modified.

PropName: Select the name of the property to be set from the dropdown list.

PropType: Select the property type to be sent: integer, floating point (string not valid)

PropValue: Select the numerical value of the property. (This must be a number. All properties have numerical equivalent values, i.e., Off=0, On=1)

ThresholdType: Selects the method of determining when to change from the OffShape and color to the OnShape and color. *(Equal*, *NotEqual*, *GreaterThan*, *LessThan*, *GreaterThanorEqual*, *LessThanorEqual)*

ThresholdValue: Sets the value that the DataPoint is compared against to determine to switch from Off to On.

DataPointShape is a dynamic element that can be assigned to an I/O value. When the I/O value changes from On to Off or crosses a threshold, the DataPointShape can change its color and/or shape to indicated a change in the I/O value. You configure an ON color and shape as well as an OFF color and shape. A change can be triggered based on a threshold or equality using the threshold type and threshold value properties.

Properties

Align: To align shape to edges of the palette area.

alBottom: Stretches the shape across the bottom of the palette.

alClient: Fills the palette with the shape. Useful for adding a background graphic.

alLeft: Stretches the shape along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Anchors: Sets one or more sides of the shape as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the element a fixed distance from the bottom of the palette.

akLeft: Anchors the element a fixed distance from the left side of the palette.

akRight: Anchors the element a fixed distance from the right side of the palette.

akTop: Anchors the element a fixed distance from the top of the palette.

Brush:

Color: Color of the shape.

Style: Pattern within the shape.

Constraints: Limits the maximum and minimum size of the shape.

MaxHeight: Sets the maximum height. (0 disables this property)

MaxWidth: Sets the maximum width. (0 disables this property)

MinHeight:Sets the minimum height. (0 disables this property)

MinWidth: Sets the minimum width. (0 disables this property)

Height: Sets the height of the shape.

Hint: Text shown on mouseover.

Left: Sets the position of the shape as measured from the left side of the palette.

OffColor: Select the color of the shape when Off.

OnColor: Select the color of the shape when On.

OnShape: Select from Circle, Ellipse, Rectangle, RoundRectangle, RoundSquare, Square.

Pen:

Color: Select the color of the line.

Mode: *pmBlack*, *pmCopy*, *pmMask*, *pmMaskNotPen*, *pmMaskPenNot*, *Merge*, *pmMergeNotPen*, *pmMergePenNot*, *pmNop*, *pmNot*, *pmNotCopy*, *pmNotMask*, *pmNotMerge*, *pmNotXor*, *pmWhite*, *pmXor*.

Style:Select from solid or various types of dotted line to represent the shape border.

Width: Select the width of the shape border.

Show Hint: Check if you want the mouseover hint to appear.

State: This is the On/Off, Enabled/Disabled Datapoint state.

ThresholdType: Selects the method of determining when to change from the OffShape and color to the OnShape and color. *(Equal*, *NotEqual*, *GreaterThan*, *LessThan*, *GreaterThanorEqual*, *LessThanorEqual)*

ThresholdValue: Sets the value that the DataPoint is compared against to determine to switch from Off to On.

Top: Sets the position of the shape as measured from the top of the palette.

Visible: Check to make the shape visible.

Width: Sets the width of the shape.

PLACING ELECTRICAL ELEMENTS ON THE PALETTE

The ELEC tab holds the dynamic graphical components related to electrical states:

Electrical Contact, Breaker, Fuse, Motor, and State Indicator.

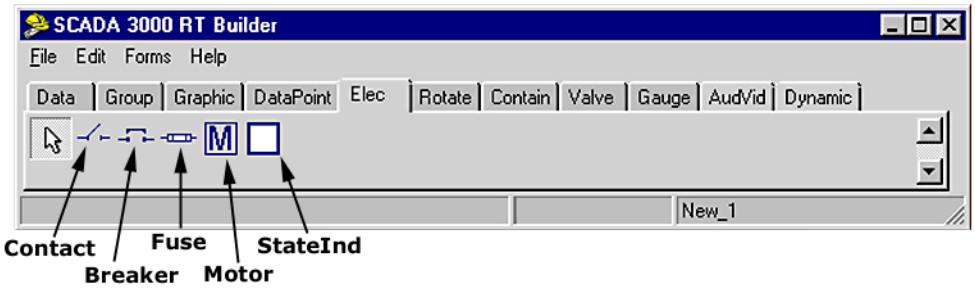


Figure 20: Electrical Tab

Properties

Align: To align graphic/component to edges of the palette area.

alBottom: Stretches the component across the bottom of the palette.

alClient: Fills the palette with the component. Useful for adding a background graphic.

alLeft: Stretches the component along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Anchors: Sets one or more sides of the component as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the component a fixed distance from the bottom of the palette.

akLeft: Anchors the component a fixed distance from the left side of the palette.

akRight: Anchors the component a fixed distance from the right side of the palette.

akTop: Anchors the component a fixed distance from the top of the palette.

BackGndMode: *bmOpaque* or *bmTransparent*.

Color: Background color, visible only if you choose *bmOpaque* above.

Constraints: Limits the maximum and minimum size of the element on the palette.

MaxHeight: Sets the maximum height. (0 disables this property)

MaxWidth: Sets the maximum width. (0 disables this property)

MinHeight: Sets the minimum height. (0 disables this property)

MinWidth: Sets the minimum width. (0 disables this property)

Datapoint: Assigns the component to an I/O point within a SCADA unit.

Datapoint: Enter the datapoint UAF or click in the *Datapoint* box above and then click the button at the end of the box to display the Datapoint selection form.

SCADA: Select the SCADA unit from which the datapoint information is coming.

Font: Sets the default font for labels and text. Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: Can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using

fpDefault as the pitch value will cause the default pitch style to be used with the font. The *fpFixed* and *fpVariable* values can be used to explicitly set the pitch; a value of *fpFixed* causes each character in the font to have the same width (monospacing), while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Vertical size in pixels (overridden if you chose a Max/Min Height).

Hint: Type in the text that will appear upon mouseover.

Left: Sets the position of the component as measured from the left side of the palette.

Off Color: Color of the component in the "off" state. Default color is Red.

On Color: Color of the component in the "on" state. Default color is Green.

Show Hint: Enables the hint message to display on mouseover.

State: Checkbox. This is the On/Off, Enabled/Disabled Datapoint state.

Status Off: Message showing if status is off. Default is (Open).

Status On: Message showing if the status is on. Default is (Closed).

Status Text: Checkbox. Check if you want the status text to display.

Stay On Top: Checkbox. Check if you want the component to float above any other graphics.

Title: Description of the component.

Visible: Checkbox. Check to make component visible.

Width: Horizontal size in pixels (overridden by Max/Min Width values, if any.)

PLACING ROTATING ELEMENTS ON THE PALETTE

There are four components on the Rotate Tab: Rotating Blower, Rotating Pump, Rotating Compressor, Rotating Turbine.

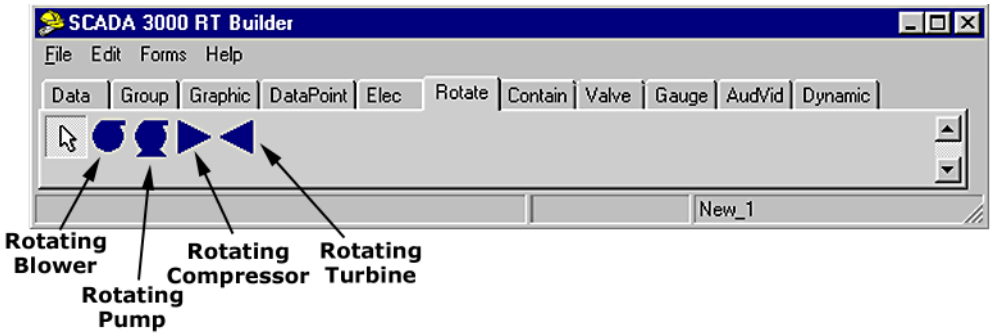


Figure 21. Rotate Tab

Properties

Align: To align graphic/component to edges of the palette area.

alBottom: Stretches the component across the bottom of the palette.

alClient: Fills the palette with the component. Useful for adding a background graphic.

alLeft: Stretches the component along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Anchors: Sets one or more sides of the element as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the element a fixed distance from the bottom of the palette.

akLeft: Anchors the element a fixed distance from the left side of the palette.

akRight: Anchors the element a fixed distance from the right side of the palette.

akTop: Anchors the element a fixed distance from the top of the palette.

BackGndMode: *bmOpaque* or *bmTransparent*.

Color: Background color, visible only if you choose *bmOpaque* above.

Constraints: Limits the maximum and minimum size of the element on the palette.

MaxHeight: Sets the maximum height. (0 disables this property)

MaxWidth: Sets the maximum width. (0 disables this property)

MinHeight: Sets the minimum height. (0 disables this property)

MinWidth: Sets the minimum width. (0 disables this property)

Datapoint: Assigns the element to an I/O point within a SCADA unit.

Datapoint: Enter the datapoint UAF or click in the *Datapoint* box above and then click the button at the end of the box to display the Datapoint selection form.

SCADA: Select the SCADA unit from which the datapoint information is coming.

Font: Sets the default font for labels and text. Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: Can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used with the font. The *fpFixed* and *fpVariable* values can be used to explicitly set the pitch; a value of *fpFixed* causes each character in the font to have the same width (monospacing), while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Vertical size in pixels (overridden if you chose a Max/Min Height).

Hint: Type in the text that will appear upon mouseover.

Left: Sets the position of the component as measured from the left side of the palette.

Off Color: Color of the component in the "off" state. Default color is Red.

On Color: Color of the component in the "on" state. Default color is Green.

Show Hint: Enables the hint message to display on mouseover.

State: Checkbox. This is the On/Off, Enabled/Disabled Datapoint state.

Stay On Top: Checkbox. Check if you want the element to float above any other graphics.

Visible: Checkbox. Check to make component visible.

Width: Horizontal size in pixels (overridden by Max/Min Width values, if any.)

PLACING CONTAINER COMPONENTS ON THE PALETTE

The Contain Tab offers the following realtime elements: ContainerTank, ContainerVessel, ContainerPipe, ContainerFill, ContainerBoiler, ContainerKettle. Most of these are simple, line-drawing representations of containers and can have dynamic properties. **Pipe**, for instance, is a configurable horizontal or vertical shape that can be linked to an I/O as an active pipe section, changing color as conditions (such as flow) change. When the assigned DataPoint value crosses the threshold value, the pipe will change to its FillColor. All of the containers include built-in bar gauges to display the fill level based on the DataPoint value. Set the MaxValue and MinValue properties to values that correspond to your tank.

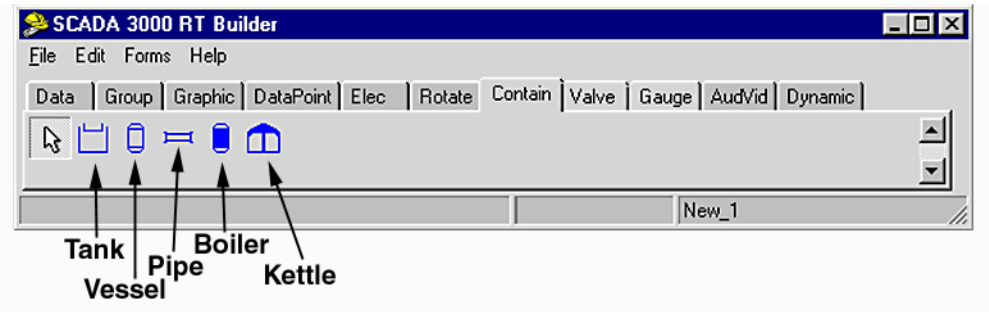


Figure 22: Contain Tab

Properties

Align: To align graphic/component to edges of the palette area.

alBottom: Stretches the component across the bottom of the palette.

alClient: Fills the palette with the component. Useful for adding a background graphic.

alLeft: Stretches the component along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Anchors: Sets one or more sides of the element as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the element a fixed distance from the bottom of the palette.

akLeft: Anchors the element a fixed distance from the left side of the palette.

akRight: Anchors the element a fixed distance from the right side of the palette.

akTop: Anchors the element a fixed distance from the top of the palette.

BackGndMode: *bmOpaque* or *bmTransparent*.

Color: Background color, visible only if you choose *bmOpaque* above.

Constraints: Limits the maximum and minimum size of the container on the palette.

MaxHeight: Sets the maximum height. (0 disables this property)

MaxWidth: Sets the maximum width. (0 disables this property)

MinHeight: Sets the minimum height. (0 disables this property)

MinWidth: Sets the minimum width. (0 disables this property)

Datapoint: Assigns the container to an I/O point within a SCADA unit.

Datapoint: Enter the datapoint UAF or click in the *Datapoint* box above and then click the button at the end of the box to display the Datapoint selection form.

SCADA: Select the SCADA unit from which the datapoint information is coming.

Display Percent (tank/vessel): Displays percent full based on Max and Min values.

Display Value (tank/vessel): Displays actual datapoint value.

Fill Color: Color of container when filled.

Fill Style: Solid or pattern fills.

Font: Sets the default font for labels and text. Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: Can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used with the font. The *fpFixed* and *fpVariable* values can be used to explicitly set the pitch; a value of *fpFixed* causes each character in the font to have the same width (monospacing), while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Vertical size in pixels (overridden if you chose a Max/Min Height).

Hint: Type in the text that will appear upon mouseover.

Left: Sets the position of the component as measured from the left side of the palette.

Max Value: Upper datapoint value limit

Min Value: Lower datapoint value limit

Show Hint: Checkbox. Enables the hint message to display on mouseover.

Stay On Top: Checkbox. Check if you want the container component to float above any other graphics.

Top: Y coordinate of the top edge of the container.

Value: Realtime datapoint value.

Visible: Checkbox. Check to make component visible.

Width: Horizontal size in pixels (overridden by Max/Min Width values, if any.)

PLACING VALVE COMPONENTS ON THE PALETTE

There are four valve components on the tab: Valve Actuator, Valve, Valve Butterfly, and Valve Meter.

Figure 23: Valve Tab

Properties

Actuator Type (Actuator only):

atAirMotor

atElecMotor

atHydraulic

atSolenoid

Align: To align graphic/component to edges of the palette area.

alBottom: Stretches the component across the bottom of the palette.

alClient: Fills the palette with the component. Useful for adding a background graphic.

alLeft: Stretches the component along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Anchors: Sets one or more sides of the element as a positional anchor with respect to the edges of the palette.

akBottom: Anchors the element a fixed distance from the bottom of the palette.

akLeft: Anchors the element a fixed distance from the left side of the palette.

akRight: Anchors the element a fixed distance from the right side of the palette.

akTop: Anchors the element a fixed distance from the top of the palette.

BackGndMode: *bmOpaque* or *bmTransparent*.

Color: Background color, visible only if you choose *bmOpaque* above.

Constraints: Limits the maximum and minimum size of the container on the palette.

MaxHeight: Sets the maximum height. (0 disables this property)

MaxWidth: Sets the maximum width. (0 disables this property)

MinHeight: Sets the minimum height. (0 disables this property)

MinWidth: Sets the minimum width. (0 disables this property)

Datapoint: Assigns the container to an I/O point within a SCADA unit.

Datapoint: Enter the datapoint UAF or click in the *Datapoint* box above and then click the button at the end of the box to display the Datapoint selection form.

SCADA: Select the SCADA unit from which the datapoint information is coming.

Fill Color (Meter): The color of the progress bar as it fills up.

Font: Sets the default font for labels and text. Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: Can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used with the font. The *fpFixed* and *fpVariable* values can be used to explicitly set the pitch; a value of *fpFixed* causes each character in the font to have the same width (monospacing), while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Vertical size in pixels (overridden if you chose a Max/Min Height).

Hint: Type in the text that will appear upon mouseover.

Left: Sets the position of the component as measured from the left side of the palette.

Max Value (Meter): Upper datapoint value limit.

Min Value (Meter): Lower datapoint value limit.

Off Color: Color of the component in the "off" state. Default color is Red.

On Color: Color of the component in the "on" state. Default color is Green.

Show Hint: Checkbox. Enables the hint message to display on mouseover.

State: Checkbox. This is the On/Off, Enabled/Disabled Datapoint state.

Stay On Top: Checkbox. Check if you want the element to float above any other graphics.

Top: Y coordinate of the top edge of the valve component.

Visible: Checkbox. Check to make component visible.

Width: Horizontal size in pixels (overridden by Max/Min Width values, if any).

PLACING GAUGES ON THE PALETTE

The Gauges tab features several bar gauges, an angular gauge, and a strip chart. The gauges can be assigned to DataPoints or analog I/O values or variables. The Strip Chart can display multiple datapoints on the same chart. (Note that the gauges and strip chart only display during Run mode.) Click on the Gauge Tab.

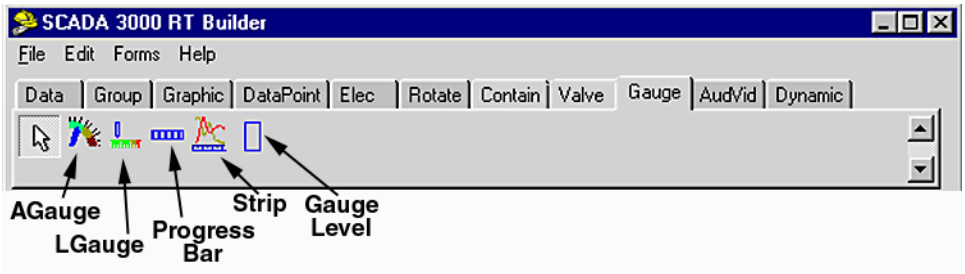


Figure 24: Gauge Tab

You have five gauge options: The **AGauge** places a dial gauge on the palette. The **LGauge** places a rectangular gauge that can be oriented either horizontally or vertically. The **ProgressBar** places a bar graph on the palette that can be used to show fills and is designed to be linked to I/Os, and stretched and resized to fit over other graphics such as containers and bitmaps. The **Strip** places a strip chart on the palette, allowing you to run a comparative chart of up to 8 I/O points. **GaugeLevel** can be placed over or beside any other bitmap image (tanks, wells) and linked to an I/O to indicate changes in levels.

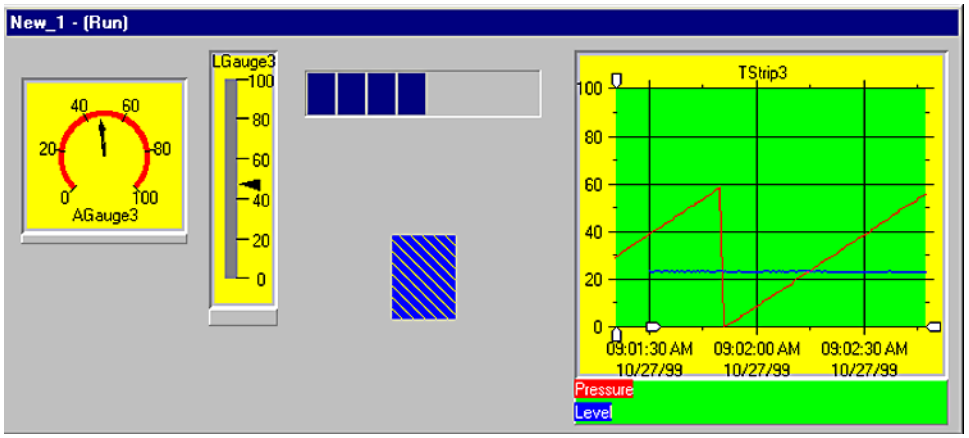


Figure 25: AGauge example

Setting Up Gauges

To use any of the gauges, simply assign the gauge to a SCADA unit and DataPoint in the properties menu. Next, enter Min and Max values for the range, and enter a title. The remaining properties can be used to tailor the gauge to your application or preferences.

Setting Up the Strip Chart

Place the strip chart on the palette and adjust its size accordingly. Expand the *DataPoint* properties and assign up to 8 datapoints to the strip chart. Enter names to describe the datapoints by

expanding the *DataNames* property. Next, change the number-of-data-points property *(NumDataPoints)* to match the number of datapoints you have entered above. Edit the title to describe your strip chart. You can further customize the look of the strip chart by editing the other strip chart properties based upon your preferences.

Properties

Align (ProgressBar/GaugeLevel): Stretches the bar to align with the top, bottom, left, right, or client part (fills all) of the palette.

Anchors (ProgressBar/GaugeLevel):

akBottom: Anchors the element a fixed distance from the bottom of the palette.

akLeft: Anchors the element a fixed distance from the left side of the palette.

akRight: Anchors the element a fixed distance from the right side of the palette.

akTop: Anchors the element a fixed distance from the top of the palette.

BackGnd Color: Choose the color of the gauge background. Sets the frame color of the strip chart.

BackGndMode (GaugeLevel): Choose either Opaque or Transparent background for the gauge when empty.

Color (GaugeLevel): Color of gauge background if Opaque is selected.

Constraints (Progress Bar/GaugeLevel): Limits the maximum and minimum size of the bar.

MaxHeight: Sets the maximum height of the bar. (0 disables this property)

MaxWidth: Sets the maximum width of the bar. (0 disables this property)

MinHeight: Sets the minimum height of the bar. (0 disables this property)

MinWidth: Sets the minimum width of the bar. (0 disables this property)

DataColors (Strip chart): Assign a different color to each of up to 8 charted I/O points.

DataFill (Strip chart): Checkbox.

DataNames (Strip chart): Assign a name to each I/O being charted.

Datapoint: Use the submenus to select the I/O (Datapoint), and the unit (SCADA) being monitored and linked to the gauge.

DataPoints (Strip Chart): Select up to eight I/O points to be charted.

DataType (Strip chart):

DataValue X: Specifies the horizontal location of the data value from -1 to 1. 0=center.

DataValue Y: Specifies the vertical location of the data value from -1 to 1. 0=center.

Display Orientation (for bar gauges): doHorizontal, and doVertical.

Display Percent (GaugeLevel): Displays percent full based on Max and Min values.

Display Value: Checkbox if you want the datapoint value displayed.

FillColor (GaugeLevel): Color of the Container or Vessel when filled.

FillDirection (GaugeLevel): Vertical or Horizontal fill representation.

FillStyle (GaugeLevel): Solid or patterned fill.

Height: Sets the height of the gauge.

Hint: Text message that appears when the cursor rests over an object.

Hub Color: select a color for the hub of the gauge.

Left: Sets the position of the gauge as measured from the left side of the palette

Legend Value (Strip chart): Shows the value of the datapoint in the legend.

Max Buffer Size: Sets the total size of the holding array buffer for the data. 100=100 datapoints.

MaxValue (ProgressBar/GaugeLevel): Upper datapoint value limit or maximum display value.

MinValue (ProgressBar/GaugeLevel): Lower datapoint value limit or minimum display value.

Needle Color: choose a color for the display needle.

Needle Style: Needle, or pointer, style options.

NumDataPoints (Strip chart): Enter the number of datapoints to be charted.

Orientation (ProgressBar): pbHorizontal, pb Vertical.

RefreshRate (Strip chart): Adjusts how often the legend values update, in seconds.

Scale Start Angle: in degrees on a circle. Default, 225, is approximately 8 o'clock on the dial.

Scale Stop Angle: in degrees on a circle. Default, 135, is approximately 4 o'clock on the dial.

Show Hint (ProgressBar/GaugeLevel): Checkbox—check to show the hint on mouseover.

Smooth (ProgressBar): Checkbox—check to smooth the bar animation.

StayOnTop (GaugeLevel): Forces the gauge to stay on top of other images when they overlap.

Step (ProgressBar):

TicAuto: Automatically formats tic marks.

TicAuto Count: Tics are used to mark intervals on the gauge's face. The count determines the number of intervals shown.

Title: Assign a name or title to a gauge.

Title X: Places the x coordinate of the title.

Title Y: Places the y coordinate of the title.

Top: Sets the position of the gauge as measured from the top of the palette.

Track Color (Strip chart): Sets the background color of the strip chart.

Value (GaugeLevel): Temporarily forces a value into the gauge to view its appearance.

Visible (Progress Bar/GaugeLevel): Checkbox, check to make the bar visible.

Width: Width of the gauge.

X Resolution (Strip chart): A decimal number which sets the scale of the x-axis based on a 24-hour period. If the X Resolution is set to a value of 1, then the x-axis will cover 24 hours. If the X Resolution is set to 0.5, the x-axis will cover 12 hours. Default = 0.001 (about 1.5 minutes).

PLACING AUDIO AND VIDEO ELEMENTS ON THE PALETTE

Two types of AV elements can be added to your palette: An AVI player for animations and movies, and a WAV player for sound files.

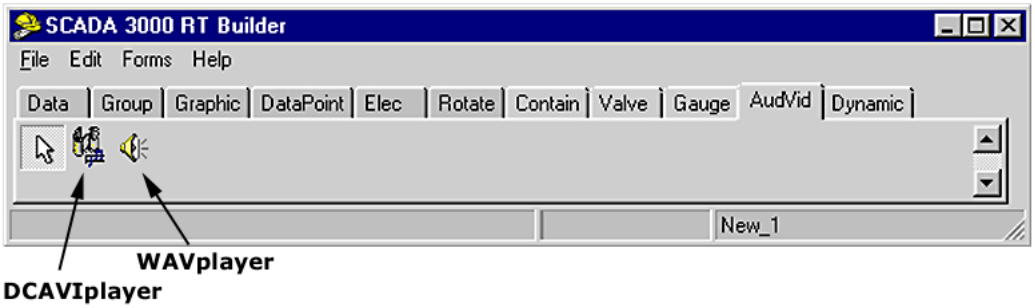


Figure 26: AudVid Tab

Playing an Audio or Video Message

You can have the real-time screen play an audio message when an I/O point meets certain conditions. The audio message must be a Windows Wave (WAV) file, which can be created with a sound card, microphone and the Windows *Sound Recorder* program located under Programs>Accessories>Multimedia. The Wave file can be set to play when an I/O value is Equal (EQ), Not Equal (NE), Less Than (LT), Less Than or Equal (LE), Greater Than (GT), or Greater Than or Equal (GE). To use this feature, follow the steps below. Note that the components placed on the palette for this feature will be invisible in Run mode.

1. Click the AudVid tab and place the Wavplayer component on the palette.
2. On the properties menu, double-click in the Filename field and select the Wave file you want played. You can click the Play box to hear the message play once.
3. Click the DataPoint tab and place the DataPointAlert component on the palette.
4. From the properties menu, set the AlertInterval. This is how often the Wave file will be played when the logical condition is true.
5. Double-click in the Datapoint field and select the I/O point you wish to use to activate the Wave file.
6. Click in the DynComponent field and click the down arrow. Choose the Waveplayer1 component.
7. Click in the DynProperty field and click the down arrow. Choose the Play property.
8. Click in the ThresholdType field and select the logical threshold you want to use to determine when to play the file.
9. Set the ThresholdValue that will be used to compare against the datapoint value.

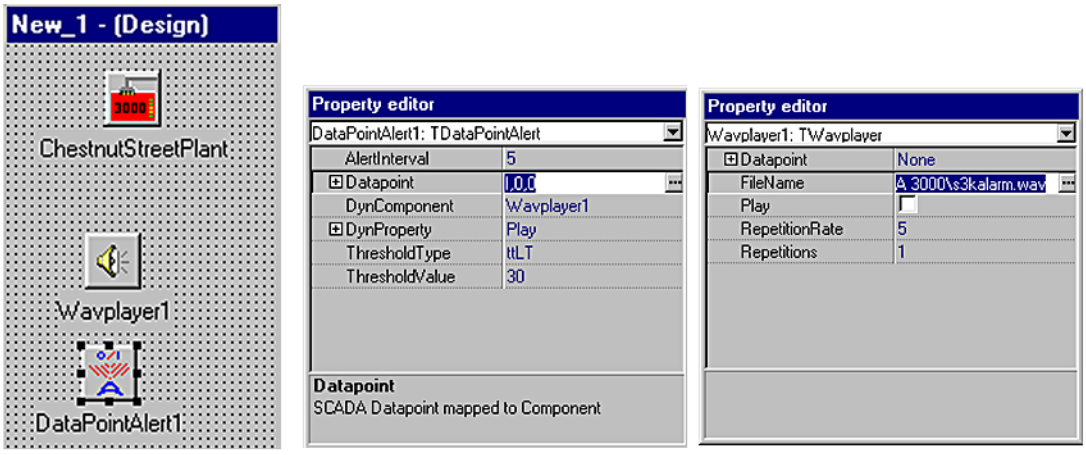


Figure 29: WAV palette & properties

Properties

Audio: Checkbox. Check to be able to hear sound.

Color: Select a color for the placeholder for the sound file.

Datapoint: Select *Datapoint* to link to a specific I/O, and *SCADA* to link to a specific RTU.

Filename: Link to the file that plays.

Play: Checkbox. Check to play the file.

Repetition Rate: How often the animation or sound file repeats.

Repetitions: How many times the file plays.

Height: Sets the height of the AV file.

Hint: Text message that appears when the cursor rests over an object.

Left: Sets the position of the gauge as measured from the left side of the palette.

Run: Checkbox. Check to have the AV file play.

Title: Assign an name or title to this AV element.

Top: Sets the position of the gauge as measured from the top of the palette.

Transparent: Checkbox. Check if you want the linked file to be transparent.

Width: Width of the AV element.

PLACING DYNAMIC ELEMENTS ON THE PALETTE

There are three dynamic component tools that can be used to perform dynamic property manipulation of other components. The function of each is described below.

The *UAF Dynamic Component* allows you to change a property of another component to the value of a datapoint. To use this option, select the datapoint to be used as the source and then select the component and property to be changed on the properties menu.

The *Dynamic Label Component* allows you to display internal elements of the strip chart, such as the maximum (Ymax) and minimum (Ymin) values of a datapoint on the strip chart (See Gauges Tab for adding a strip chart). To use this feature, select the Strip Chart from the DynComponent list, and then select the element to be displayed from the DynProperty list.

DynAlert allows you to monitor the property value of one component (source) and to change the property of another component (destination) when the source value is logically compared to a threshold (equal to, less than, greater than, etc.) and is found to be true.

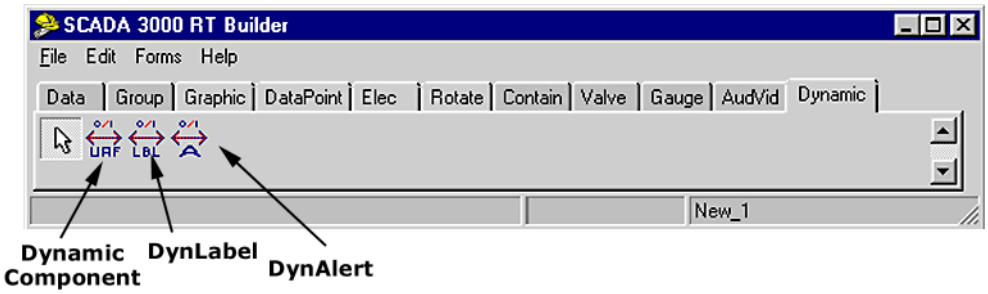


Figure 27: Dynamic Tab

Properties

Alert Interval (DynAlert): default=1

Align: To align graphic/component to edges of the palette area.

alBottom: Stretches the component across the bottom of the palette.

alClient: Fills the palette with the component. Useful for adding a background graphic.

alLeft: Stretches the component along the left edge of the palette.

alNone: No realignment is applied. *Note:* this will not correct any other previously applied alignment.

alRight: Stretches the component along the right edge of the palette.

alTop: Stretches the component across the top of the palette.

Alignment: *taCenter*, *taLeftJustify*, *taRightJustify*. Aligns the text in the dynamic area.

Autosize: Checkbox

Caption: Type in any custom caption to go with the dynamic element.

Color: Color of the component.

Datapoint: Assigns the container to an I/O point within a SCADA unit.

Datapoint: Enter the datapoint UAF or click in the *Datapoint* box above and then click the button at the end of the box to display the Datapoint selection form.

SCADA: Select the SCADA unit from which the datapoint information is coming.

DynComponent (DynLabel): Component to which Datapoint is mapped.

DynInputComponent (DynAlert): *DynComp1*, *DynAlert1*

DynInputProperty (DynAlert): default=none

PropName: Type a name to identify the input property.

PropType: *dpFloat*, *dpInteger*, *dpString*

PropValue: default=0

DynOutputComponent (DynAlert): *DynComp1*, *DynAlert1*

DynOutputProperty (DynAlert): default=none

PropName: type a name to identify the output property.

PropType: *dpFloat*, *dpInteger*, *dpString*

PropValue: default=0

Font: Sets the default font for labels and text. Click in the font box, then click the small button at the end of the box to select a font.

Charset: Specifies the character set of the font.

Color: Sets the color of the text.

Height: Sets the height of the font object in pixels. If you wish to use a specific point size, you should set the size property.

Pitch: Can be set to one of three values: *fpDefault*, *fpFixed*, or *fpVariable*. Using *fpDefault* as the pitch value will cause the default pitch style to be used with the font. The *fpFixed* and *fpVariable* values can be used to explicitly set the pitch; a value of *fpFixed* causes each character in the font to have the same width (monospacing), while a value of *fpVariable* allows different characters within a font to have unique widths.

Size: Sets the point size of a font object. If you need to ensure that the font fits within a specified area, you can manipulate the Height and Pitch properties.

Style: Sets the default style of the font or text.

fsBold: Sets the default to boldface.

fsItalic: Sets the default to italic.

fsStrikeout: Forces a line through the font.

fsUnderline: Underlines the text as default.

Height: Vertical size in pixels (overridden if you chose a Max/Min Height).

Hint: Type in the text that will appear upon mouseover.

Layout:

Left: Sets the position of the component as measured from the left side of the palette.

Show Hint: Checkbox. Enables the hint message to display on mouseover.

Threshold Type (DynAlert): *ttEQ*, *ttGE*, *ttGT*, *ttLE*, *ttLT*, *ttNE*

Threshold Value (DynAlert): default=0

Title: Description of the component.

Top: Sets the position of the component as measured from the top of the palette.

Transparent: Checkbox. Check if you want the linked file to be transparent.

Visible: Checkbox. Check to make the element visible.

Width: Width of the Dynamic Component.

Word Wrap (Label): Checkbox. Check to have text wrap automatically in the label space.

REAL-TIME OUTPUT CONTROL

Outputs can be controlled from the real-time screen by placing components on the palette and assigning them to Output datapoints. To control the Output, right-click on the component and the Force Datapoint form will appear. This form allows you to force an output on or off, and also to unforce an output. The Force Datapoint form is shown below:

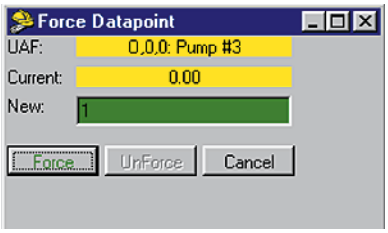


Figure 28: Force Datapoint form

The Force Datapoint form shows the datapoint UAF, the label associated with that UAF, and the present value. To turn a digital output ON, type *1* in the *New* value field and click the *Force* button. To turn a digital output OFF, type *0* in the *New* value field and click the *Force* button. To unforce an output, click the *Unforce* button. Click *Cancel* to exit the form without making any changes. Only components which allow a datapoint to be assigned can be used to control an output.

SAVING THE REAL-TIME SCREEN

To save the Real-time screen, select *File*, then *Save* from the RT Builder main menu. Enter a file name for your screen and click OK. The screen layout with all of its components, labels, and graphics will be saved.

LOADING A REAL-TIME SCREEN FOR EDITING

To edit a previously saved Real-time screen, select *File* and then *Load* from the RT Builder main menu. Select the real-time screen file name you wish to edit, then click OK.

RUNNING THE REAL-TIME SCREEN

Running the Real-time screen will make the SCADA 3000 software log in to each unit specified on the screen and begin updating the values of the components on the screen. The update rate will depend on how many SCADA units the software must log into, what type of communication is used (radio, telephone, serial port), and the number of components on the screen. SCADA units that must be contacted via telephone will be updated one at a time based on the connect time parameters programmed for those units. When connecting to units via radio be sure that your radio network has already been thoroughly tested and found to be satisfactory. Don't use the realtime screen as your first attempt at radio communications.

To start the Real-time screen, click on the *R.T.* button from the Sensaphone Control Center form, or select *Real-Time* from the main menu and choose *Run*. Next, select the file name of the real-time screen you wish to run, and click OK. Note that you can maximize the Real-time screen by double-clicking on the bar across the top of the screen. To exit from the Real-time screen, select *File*, then *Close* from the main menu of the R.T. Builder form.

Chapter 15: LADDER LOGIC PROGRAMMING

Ladder Logic is one of the languages used to program the SCADA 3000 for control functions. While the C language is more mathematically capable, it does not handle "On/Off" functions very well. That's where Ladder Logic comes in. It is the preferred language for control functions. With C Programming, you write formulas. With Ladder Logic, you build constructs out of a set of **symbols**. The ladder symbols and expressions represent the program logic used to control the machine or process. By putting together the various symbols you can build very sophisticated control programs. The rules/conventions that dictate how programs can be constructed are listed within this section. Listed below are the ladder elements and functions that are supported in the SCADA 3000. A description of each follows the list. Example programs are illustrated at the end of this chapter.

Note: Windows NT is required if you plan on writing programs that contain 500 or more instructions.

Ladder Logic Elements Supported

XIC	Examine if Closed
XIO	Examine if Open
OTE	Output energize
OTL	Output latch
OTU	Output Unlatch
OSR	One shot rising
TON	Timer on-delay
TOF	Timer off-delay
RTO	Retentive timer
CTU	Count up counter
CTD	Count down counter
RES	Reset timer or counter
MOV	Move

Comparison Instructions

EQU	Equal
NEQ	Not Equal
LES	Less than
LEQ	Less than or equal
GRT	Greater than
GEQ	Greater than or equal

Math Instructions

ADD	Addition
SUB	Subtraction
MUL	Multiplication
DIV	Division
POW	Power

Rules/Conventions of Ladder Logic Programming

1) One output per rung.

2) Each rung controls an output.

3) A normally open contact (XIC Instruction) when evaluated by the program is examined for a 1 (closed contact) to make the instruction true.

4) A normally closed contact (XIO Instruction) when evaluated by the program is examined for a 0 (Open contact) to make the instruction true.

5) The output on a given rung will be energized if any left to right path has all contacts closed.

6) An output can control either a connected device (if the UAF is also a termination point) or an internal output used exclusively within the program.

7) An internal output does not control a field device. It provides interlocking functions within the program.

8) Input contacts can represent input signals sent from connected inputs, contacts from internal outputs, or contacts from connected outputs.

9) Repeated use of contacts is allowed. A given input, output or internal output can be used throughout the program as many times as required.

10) Contacts can be programmed in series or parallel depending on the output control logic required.

11) The maximum number of parallel branches is limited to 6.

12) Only 1 Counter or timer per rung.

13) A RES instruction resets all status bits and the accumulated value for the counter or timer that is being addressed.

14) Do not use a RES instruction on a TOF instruction.

15) The OSR instruction is not allowed inside a branch.

16) Preset and Accumulated values for timers range from 0 to +32767. If a timer preset or accumulated value is a negative number, a runtime error will occur.

17) The timer timebase is fixed at 0.01 seconds (10ms).

18) The preset and accumulated values for counters range from -32768 to +32767, and are stored as signed integers. Negative values are stored in 2's complement form.

19) When using an RES instruction to reset an alarm, an OSR instruction should be used preceding the RES to prevent continuous alarm resetting.

Description of Ladder Elements and Functions

Each ladder logic symbol or functional block will contain an address field, a title field and a comment field. Timers and counters will contain additional fields for preset's and accumulated values. Listed below is the logic definition for the SCADA 3000.

Logic Definition

Contact State	Bit Value	Input Voltage
Open	0 (Off)	5V
Closed	1 (On)	0V

XIC Instruction (Examine if Closed)

This instruction is used to determine if a contact has closed or to check if a bit is ON. When the address represents a discrete input, the program checks the open/closed status of the contact connected to the input. If the contact is closed, then the instruction is evaluated as true.When the address represents a bit value of an instruction, the program checks to see if the bit is ON. If the bit is on, then the instruction is evaluated as true.

The instruction executes in 2.8µs and requires 24 bytes.

Following is a list of the data points that may be used with XIC instructions:

- Inputs, when configured as a contact closure
- Relay outputs
- Timer Status bits (Done, Running, Enable)
- Counter Status bits (Done, Count-up enable, Count-down enable, Overflow, Underflow)
- Ladder bits
- C variables, when used as bit values (0 or 1)
- Gas Flow (enable only)

XIO Instruction (Examine if Open)

This instruction is used to determine if a contact has opened or to check if a bit is OFF. When the address represents a discrete input, the program checks the open/closed status of the contact connected to the input. If the contact is open, then the instruction is evaluated as true. When the address represents a bit value of an instruction, the program checks to see if the bit is OFF. If the bit is OFF, then the instruction is evaluated as true.

The instruction executes in 2.8µs and requires 24 bytes.

Following is a list of the data points that may be used with XIO instructions:

- Inputs, when configured as a contact closure
- Relay outputs
- Timer Status bits (Done, Running, Enable)
- Counter Status bits (Done, Count-up enable, Count-down enable, Overflow, Underflow)
- Ladder bits
- C variables, when used as bit values (0 or 1)
- Gas Flow (enable only)

OTE Instruction (Output Energize)

This instruction is used to turn on an output or a bit when rung conditions are true. A typical output would consist of a relay or an indicator light. OTE instructions are reset when rung conditions become false or the program is stopped and restarted.

The OTE instruction executes in 3.1µs and requires 38 bytes.

Following is a list of the data points that may be used with OTE instructions:

Relay outputs
Ladder bits

OTL Instruction (Output Latch)

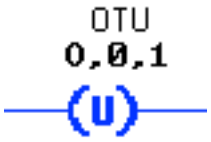
This instruction is used to latch an output or a bit when rung conditions are true. The output will remain latched even if rung conditions become false. A latched output (OTL) can only be turned off using the OTU (Output unlatch) instruction.

The OTL instruction executes in 3.1µs and requires 42 bytes.

Following is a list of the data points that may be used with OTL instructions:

Relay outputs
Ladder bits

OTU (Output Unlatch)



This instruction is used to turn off a latched output or bit when rung conditions are true. Thereafter, the output or bit remains off, regardless of the rung condition, until it is turned on (typically by an OTL instruction in another rung). The instruction executes in 3.1µs and requires 42 bytes.

Following is a list of the data points that may be used with OTU instructions:

Relay outputs
Ladder bits

OSR (One-Shot Rising)

This instruction is used to make an event occur one time. It is not associated with a particular input but rather with the instructions preceding it on the rung itself. When rung conditions preceding the OSR instruction go from false-to-true, the OSR instruction will be true for one scan. After one scan is complete, the OSR instruction becomes false, even if the rung conditions preceding it remain true. The OSR instruction will only become true again if the rung conditions preceding it transition from false-to-true again.

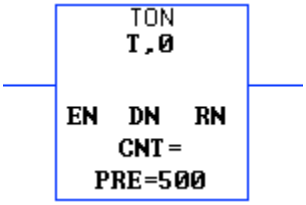
The OSR address identifies the location where the OSR bit is stored. The OSR bit is used to remember the previous rung state. The bit address you use for this instruction must be unique. The instruction executes in 3.5µs and requires 44 bytes.

Following is a list of the data points that may be used with OSR instructions:

Ladder bits

Note: You can only use **one** OSR instruction per rung, and you cannot use the OSR instruction in a branch.

TON (Timer On-Delay)



This instruction is used to turn an output ON or OFF after the timer has been on for a preset time interval. The specified timer starts at 0 and begins running when the rung status changes from false to true. It then keeps running, incrementing the TIMER VALUE, as long as the rung remains true, until it reaches the PRESET value. If the rung becomes false, then the timer stops and resets to 0. The timebase for the TON instruction is .01 seconds (for example PRESET = 500 is 5 seconds). You can have up to 64 timers. The timer bits are updated as follows:

ENABLED: Set when rung is true, and reset when rung is false.

RUNNING: Set when rung is true and TIMER VALUE is less than PRESET, and reset when rung is false or when DONE bit is set.

DONE: Set when TIMER VALUE is equal to or greater than PRESET, and reset when rung is false.

The timer address can only reference one of the 64 internal timers T,0 through T,63. Right-click on the instruction and choose *Select Timer*. A list of the 64 timers will be displayed. Select one and click *OK*. Set the Preset by right-clicking on the instruction and choosing *Set Preset*. Enter the preset value and click *OK*. The instruction executes in 4.5µs and requires 84 bytes.

The GROUP for this type is the timer number from 0-63. The ELEMENT specifies an individual function for the selected timer. There are 5 functions associated with each timer:

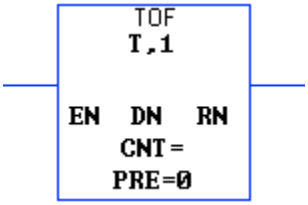
Element Number	Function Description	Range of Values
0	Timer Enabled	0 or 1 (0=disabled, 1=enabled)
1	Timer Running	0 or 1 (0=disabled, 1=enabled)
2	Timer Done	0 or 1 (0=disabled, 1=enabled)
3	Preset Value	0 to 32,767
4	Timer Value	0 to 32,767

Examples: T,0,0 indicates if Timer 0 is enabled or disabled.

T, 12,4 references the current timer value of timer number 12.

The timer value and status bits can be viewed or changed from the *Timer Status* form. To view the timer values, click on the *Ladder and C Variables* button from the Toolbar on the Main Programming form. Select the *Timers* tab. In addition to viewing the state of the timer values, you can also force changes by double-clicking on the timer value you want to change. A *Change* form will appear to let you enter a new value for the timer. Use the Up/Down arrows to flip through the pages to access the timers you wish to view.

TOF (Timer Off-Delay)



This instruction is used to turn an output ON or OFF after the rung has been false for a preset time interval. The specified timer starts running when the rung status changes from true to false. The timer starts at 0 and counts up to the PRESET value. The TIMER VALUE will continue to increment as long as the rung remains false, until it reaches the PRESET value. If the rung becomes true, then the timer stops and resets to 0, regardless of whether the timer has timed-out. The timebase for the TOF instruction is .01 seconds (for example PRESET = 500 is 5 seconds). The timer bits are updated as follows:

ENABLED: Set when rung is true, and reset when rung becomes false.

RUNNING: Set when rung is false and the TIMER VALUE is less than PRESET value, and reset when rung is true or when DONE bit is reset.

DONE: Set when rung is true, and reset when rung becomes false and the TIMER VALUE is greater than or equal to the PRESET value.

The timer address can only reference one of the 64 internal timers T,0 through T,63. Right-click on the instruction and choose *Select Timer*. A list of the 64 timers will be displayed. Select one and click *OK*. Set the Preset by right-clicking on the instruction and choosing *Set Preset*. Enter the preset value and click *OK*.

The instruction executes in 4.5µs and requires 84 bytes.

NOTE: The RESET (RES) instruction cannot be used with the TOF instruction.

The GROUP for this type is the timer number from 0-63. The ELEMENT specifies an individual function for the selected timer. There are 5 functions associated with each timer:

Element Number	Function Description	Range of Values
0	Timer Enabled	0 or 1 (0=disabled, 1=enabled)

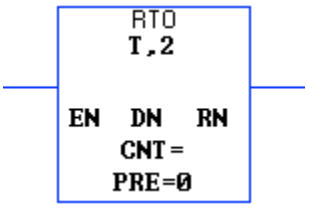
1	Timer Running	0 or 1 (0=disabled, 1=enabled)
2	Timer Done	0 or 1 (0=disabled, 1=enabled)
3	Preset Value	0 to 32,767
4	Timer Value	0 to 32,767

Examples: T,0,0 indicates if Timer 0 is enabled or disabled.

T, 12,4 references the current timer value of timer number 12.

The timer value and status bits can be viewed or changed from the *Timer Status* form. To view the timer values, click on the *Ladder and C Variables* button from the Toolbar on the Main Programming form. Select the *Timers* tab. In addition to viewing the state of the timer values, you can also force changes by double-clicking on the timer value you want to change. A *Change* form will appear to let you enter a new value for the timer. Use the Up/Down arrows to flip through the pages to access the timers you wish to view.

RTO (Retentive Timer)



This instruction is used to turn an output ON or OFF after the timer has been on for a preset time interval. The specified timer starts at 0 and increments whenever the rung status is true. It stops when the rung status is false, but it does not reset the TIMER VALUE as in the timer ON/OFF instructions. The timer can only be reset to zero from the RES(RESET) ladder instruction having the same address as the timer. The timebase for the RTO instruction is .01 seconds (for example PRESET = 500 is 5 seconds). The timer bits are updated as follows:

ENABLED: Set when rung is true, and reset when rung becomes false.

RUNNING: Set when rung is true and TIMER VALUE is less than PRESET, and reset when rung is false or when DONE bit is set.

DONE: Set when TIMER VALUE is equal to or greater than PRESET, and reset when rung is false.

The timer address can only reference one of the 64 internal timers T,0 through T,63. Right-click on the instruction and choose *Select Timer*. A list of the 64 timers will be displayed. Select one and click *OK*. Set the preset by right-clicking on the instruction and choosing *Set Preset*. Enter the preset value and click *OK*.

The instruction executes in 4.5µs and requires 84 bytes.

The GROUP for this type is the timer number from 0-63. The ELEMENT specifies an individual function for the selected timer. There are 5 functions associated with each timer:

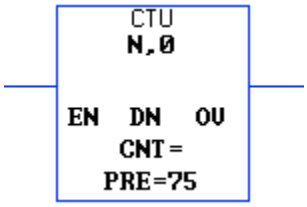
Element Number	**Function Description**	**Range of Values**
0	Timer Enabled	0 or 1 (0=disabled, 1=enabled)
1	Timer Running	0 or 1 (0=disabled, 1=enabled)
2	Timer Done	0 or 1 (0=disabled, 1=enabled)
3	Preset Value	0 to 32,767
4	Timer Value	0 to 32,767

Examples: T,0,0 indicates if Timer 0 is enabled or disabled.

T, 12,4 references the current timer value of timer number 12.

The timer value and status bits can be viewed or changed from the *Timer Status* form. To view the timer values, click on the *Ladder and C Variables* button from the Toolbar on the Main Programming form. Select the *Timers* tab. In addition to viewing the state of the timer values, you can also force changes by double-clicking on the timer value you want to change. A *Change* form will appear to let you enter a new value for the timer. Use the Up/Down arrows to flip through the pages to access the timers you wish to view.

CTU (Count-Up Counter)



This instruction counts false-to-true rung transitions starting with a COUNT value of 0. Rung transitions can be caused by events occurring in the program (internal logic) or by external field devices. When rung conditions for a CTU instruction have made a false-to-true transition, the COUNT value is incremented by one, provided that the rung containing the CTU instruction is evaluated between these transitions. When the COUNT value reaches the PRESET, the DONE bit is set. The ability of the counter to detect false-to-true transitions depends on the speed (frequency) of the incoming signal (see ladder timing specifications). The COUNT value is retained when the rung conditions again become false. The COUNT can only be set to zero using the RES(RESET) instruction that has the same address as the counter. The counter overflows at 32767, but will continue counting from -32768 if not reset. The counter bits are updated as follows:

ENABLED: Set when rung is true, and reset when rung is false. Also reset from a RES(RESET) ladder instruction.

DONE: Set when COUNT is equal to or greater than PRESET. Also reset from a RES(RESET) ladder instruction.

OVERFLOW: Set when the COUNT increments past 32767.

The CTU address identifies the register where the COUNT value is stored. The counter address can only reference one of the 64 internal counters: N,0 through N,63. Right-click on the instruction and choose *Select Counter*. A list of the 64 counters will be displayed. Select one and click *OK*. Next, set the preset by right-clicking on the instruction and choosing *Set Preset*. Enter the preset value and click *OK*. You may use the CTU instruction in combination with the CTD function (eg. One increments the count and the other decrements the count) by using the same address. In this application the COUNT Value and the PRESET value for each instruction are one in the same.

The instruction executes in 5.7µs and requires 176 bytes.

The GROUP for this type is the counter number from 0-63. The ELEMENT specifies an individual function for the selected counter. There are 7 functions associated with each timer:

Element Number	Function Description	Range of Values
0	Up Counter Enabled	0 or 1 (0=disabled, 1=enabled)
1	Down Counter Enabled	0 or 1 (0=disabled, 1=enabled)

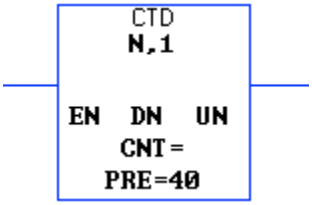
2	Counter Done	0 or 1 (0=disabled, 1=enabled)
3	Counter Overflow	0 or 1 (0=disabled, 1=enabled)
4	Counter Underflow	0 or 1 (0=disabled, 1=enabled)
5	Counter Preset	-32,768 to 32,767
6	Counter Value	-32,768 to 32,767

Examples: C,0,0 This UAF indicates if Counter 0 is enabled or disabled.

C, 19,7 References the current value of Counter number 19.

The counter value and status bits can be viewed or changed from the *Counter Status* form. To view the counter values, click on the *Ladder and C Variables* button from the Toolbar on the Main Programming form. Select the *Counters* tab. In addition to viewing the state of the counter values, you can also force changes by double-clicking on the counter value you want to change. A *Change* form will appear to let you enter a new value for the counter. Use the Up/Down arrows to flip through the pages to access the counters you wish to view.

CTD (Count-Down Counter)



This instruction counts down false-to-true rung transitions starting from 0. Rung transitions can be caused by events occurring in the program (internal logic) or by external field devices. When rung conditions for a CTD instruction have made a false-to-true transition, the COUNT value is deccremented by one, provided that the rung containing the CTD instruction is evaluated between these transitions. The ability of the counter to detect false-to-true transitions depends on the speed (frequency) of the incoming signal (see ladder timing specifications). The COUNT value is retained when the rung conditions again become false. The COUNT can only be set to zero using the RES(RESET) instruction that has the same address as the counter. The counter underflows at -32768, but will continue counting from 32767 if not reset. The counter bits are updated as follows:

ENABLED: Set when rung is true, and reset when rung is false. Also reset from a RES(RESET) ladder instruction.

DONE: Set when the COUNT is greater than or equal to the PRESET value. Reset when COUNT is less than PRESET.

UNDERFLOW: Set when the COUNT decrements below -32768. Resets when the COUNT is incremented greater than or equal to +32768 with a CTU instruction or by the RES(RESET) instruction.

The CTD address identifies the register where the COUNT value is stored. The counter address can only reference one of the 64 internal counters: N,0 through N,63. Right-click on the instruction and choose *Select Counter*. A list of the 64 counters will be displayed. Select one and click *OK*. Next, set the preset by right-clicking on the instruction and choosing *Set Preset*. Enter the preset value and click *OK*. You may use the CTD instruction in combination with the CTU function (eg. One increments the count and the other decrements the count) by using the same address. In this application the COUNT Value and the PRESET value for each instruction are one in the same.

The instruction executes in 5.7µs and requires 176 bytes.

The GROUP for this type is the counter number from 0-63. The ELEMENT specifies an individual function for the selected counter. There are 7 functions associated with each timer:

Element Number	Function Description	Range of Values
0	Up Counter Enabled	0 or 1 (0=disabled, 1=enabled)
1	Down Counter Enabled	0 or 1 (0=disabled, 1=enabled)
2	Counter Done	0 or 1 (0=disabled, 1=enabled)
3	Counter Overflow	0 or 1 (0=disabled, 1=enabled)
4	Counter Underflow	0 or 1 (0=disabled, 1=enabled)
5	Counter Preset	-32,768 to 32,767
6	Counter Value	-32,768 to 32,767

Examples: C,0,0 This UAF indicates if Counter 0 is enabled or disabled.

C, 19,7 References the current value of Counter number 19.

The counter value and status bits can be viewed or changed from the *Counter Status* form. To view the counter values, click on the *Ladder and C Variables* button from the Toolbar on the Main Programming form. Select the *Counters* tab. In addition to viewing the state of the counter values, you can also force changes by double-clicking on the counter value you want to change. A *Change* form will appear to let you enter a new value for the counter. Use the Up/Down arrows to flip through the pages to access the counters you wish to view.

RES (RESET)

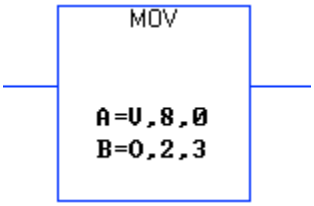
This instruction is used to reset a timer, counter, or output. You can also reset an alarm bit which is equivalent to acknowledging an alarm. When rung conditions become true the RES instruction will reset the Timer On Delay (TON), Retentive Timer (RTO), Count Up (CTU), or Count Down (CTD) instruction having the same address as the RES instruction. Counter and Timer values will be reset to 0. If the counter preset value is negative, the done bit will be set by a RES instruction.

The instruction executes in 2.5µs and requires 22 bytes.

Following is a list of the data points that may be used with RES instructions:

Outputs
Timer values
Counter values
Alarms

MOV (Move)



The MOV instruction moves the value in the memory location referenced by the Source to the destination memory location when rung conditions are true. The source value can be a fixed constant, a value contained in I/O, or a variable stored in any memory location.

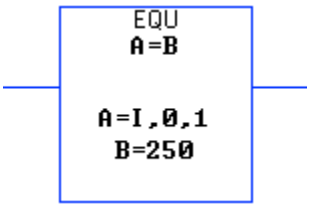
The MOV instruction executes in 2.9µs and requires 44 bytes.

Valid Sources	Valid Destinations
Inputs	Outputs
Outputs	Ladder Variables
Timers (preset & value)	C Variables
Counters (preset & value)	
Ladder Variables	
C Variables	
Gas Flow values (enable and flow rate only)	
Diagnostics	
Alarms	

COMPARISON INSTRUCTIONS

Comparison instructions are used to test two values to condition the logical continuity of a rung. Note that all values used in comparison instructions will be floating point numbers. There is no output destination, if the statement is true then rung status is continued.

EQU (EQUAL)



This instruction will compare the value of Source A to Source B, to see if they are equal. If they are equal then the rung is evaluated as true, otherwise the rung is false. Source A must be an address. Source B can either be a program constant (integer only) or an address.

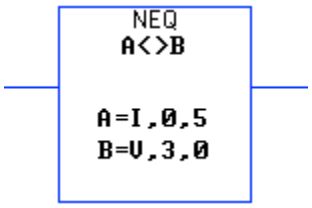
The EQU instruction executes in 15µs and requires 44 bytes.

Following is a list of the data points that may be used with EQU instructions:

- Inputs
- Outputs
- Timers (preset & value)
- Counters (preset & value)
- Ladder variables

C variables
Gas flow values (enable and flow rate only)
Diagnostics
Alarms

NEQ (NOT EQUAL)



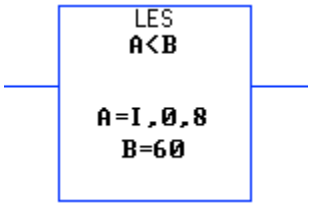
This instruction will compare the value of Source A to Source B, to see if they are NOT equal. If they are NOT equal then the rung is evaluated as true, otherwise the rung is false. Source A must be an address. Source B can either be a program constant (integer only) or an address.

The NEQ instruction executes in 15µs and requires 44 bytes.

Following is a list of the data points that may be used with NEQ instructions:

Inputs
Outputs
Timers (preset & value)
Counters (preset & value)
Ladder variables
C variables
Gas flow values (enable and flow rate only)
Diagnostics
Alarms

LES (LESS THAN)



This instruction will test to see if Source A is less than Source B. If Source A is less than Source B then the rung is evaluated as true, otherwise the rung is false. Source A must be an address. Source B can either be a program constant (integer only) or an address.

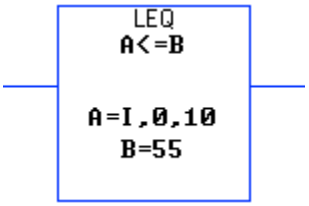
The LES instruction executes in 15µs and requires 44 bytes.

Following is a list of the data points that may be used with LES instructions:

Inputs
Outputs
Timers (preset & value)
Counters (preset & value)
Ladder variables

C variables
Gas flow values (enable and flow rate only)
Diagnostics
Alarms

LEQ (LESS THAN OR EQUAL)



This instruction will test to see if Source A is less than or equal to Source B. If Source A is less than or equal to Source B then the rung is evaluated as true, otherwise the rung is false. Source A must be an address. Source B can either be a program constant (integer only) or an address.

The LEQ instruction executes in 15µs and requires 44 bytes.

Following is a list of the data points that may be used with LEQ instructions:

Inputs
Outputs
Timers (preset & value)
Counters (preset & value)
Ladder variables
C variables
Gas flow values (enable and flow rate only)
Diagnostics
Alarms

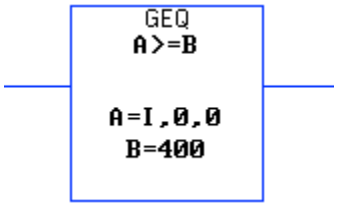
GRT (GREATER THAN)

This instruction will test to see if Source A is greater than Source B. If Source A is greater than Source B then the rung is evaluated as true, otherwise the rung is false. Source A must be an address. Source B can either be a program constant (integer only) or an address.
The GRT instruction executes in 15µs and requires 44 bytes.
Following is a list of the data points that may be used with GRT instructions:

Inputs
Outputs
Timers (preset & value)
Counters (preset & value)
Ladder variables
C variables

Gas flow values (enable and flow rate only)
Diagnostics
Alarms

GEQ (GREATER THAN OR EQUAL)



This instruction will test to see if Source A is greater than or equal to Source B. If Source A is greater than or equal to Source B then the rung is evaluated as true, otherwise the rung is false. Source A must be an address. Source B can either be a program constant (integer only) or an address.

The GEQ instruction executes in 15µs and requires 44 bytes.

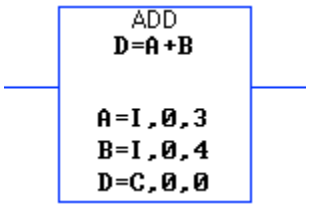
Following is a list of the data points that may be used with GEQ instructions:

Inputs
Outputs
Timers (preset & value)
Counters (preset & value)
Ladder variables
C variables
Gas flow values (enable and flow rate only)
Diagnostics
Alarms

MATH INSTRUCTIONS

The SCADA 3000 incorporates basic math instructions including: add, subtract, multiply, divide and power. These instructions will take two input values, perform the specified arithmetic function, and output the result to the specified address location.

ADD (Addition)



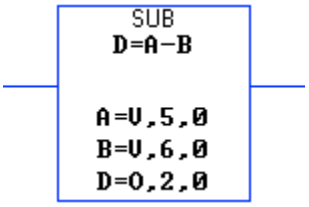
The ADD instruction adds the values stored in the two referenced memory locations, labeled as Source A and Source B when rung conditions are true. The result is stored in the destination memory location. The operand values can be fixed constants, values contained in I/O, or variables stored in any memory location.

The ADD instruction executes in 15µs and requires 68 bytes.

Following is a list of the data points that may be used with ADD instructions:

Valid Sources	Valid Destinations
Inputs	Outputs
Outputs	Ladder variables
Timers (preset & value)	C Variables
Counters (preset & value)	
Ladder variables	
C variables	
Gas flow values (enable and flow rate only)	
Diagnostics	
Alarms	

SUB (Subtraction)



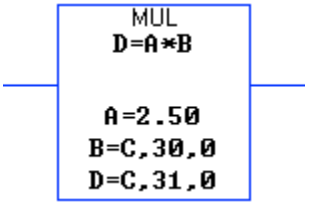
The SUB instruction subtracts the value stored in the memory location referenced by Source B from the value stored in the memory location referenced by Source A when rung conditions are true. The result is stored in the destination memory location. The operand values can be fixed constants, values contained in I/O, or variables stored in any memory location.

The SUB instruction executes in 15µs and requires 58 bytes.

Following is a list of the data points that may be used with SUB instructions:

Valid Sources	Valid Destinations
Inputs	Outputs
Outputs	Ladder variables
Timers (preset & value)	C Variables
Counters (preset & value)	
Ladder variables	
C variables	
Gas flow values (enable and flow rate only)	
Diagnostics	
Alarms	

MULT (Multiplication)



The MULT instruction multiplies the values stored in the two referenced memory locations, labeled as Source A and Source B when rung conditions are true. The result is stored in the

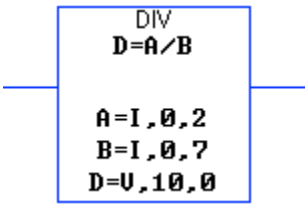
destination memory location. The operand values can be fixed constants, values contained in I/O, or variables stored in any memory location.

The MULT instruction executes in 15µs and requires 68 bytes.

Following is a list of the data points that may be used with MULT instructions:

Valid Sources	Valid Destinations
Inputs	Outputs
Outputs	Ladder variables
Timers (preset & value)	C Variables
Counters (preset & value)	
Ladder variables	
C variables	
Gas flow values (enable and flow rate only)	
Diagnostics	
Alarms	

DIV (Division)



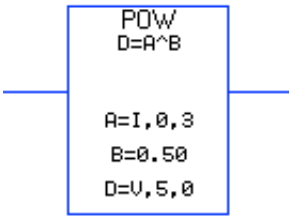
The DIV instruction divides the value stored in the memory location referenced by Source A by the value stored in the memory location referenced by Source B when rung conditions are true. The result is stored in the destination memory location. The operand values can be fixed constants, values contained in I/O, or variables stored in any memory location.

The DIV instruction executes in 15µs and requires 80 bytes.

Following is a list of the data points that may be used with DIV instructions:

Valid Sources	Valid Destinations
Inputs	Outputs
Outputs	Ladder variables
Timers (preset & value)	C Variables
Counters (preset & value)	
Ladder variables	
C variables	
Gas flow values (enable and flow rate only)	
Diagnostics	
Alarms	

POW (A to the Power of B)



The Power instruction raises the value stored in the memory location referenced by Source A by the value stored in the memory location referenced by Source B when rung conditions are true. The result is stored in the destination memory location D. The operand values can be fixed constants, values contained in I/O, or variables stored in any memory location.

The POW instruction executes in 45µs and requires 114 bytes.

Valid Sources	Valid Destinations
Inputs	Outputs
Outputs	Ladder variables
Timers (preset & value)	C Variables
Counters (preset & value)	
Ladder variables	
C variables	
Gas flow values (flow rate only)	
Diagnostics	
Alarms	

Ladder Logic Program Development

The SCADA 3000 Ladder Logic editor allows you to build, save, and load complex ladder programs for use with the SCADA 3000 main unit and expansion modules. The ladder program can run by itself or in combination with the C-Program. Specifications of the Ladder Program include:

- 32K of Program memory (about 1000 instructions—note that Windows NT is required if you plan to write programs of 500 or more instructions)
- 64 Timers
- 64 Counters
- 128 Floating Point variables
- 4096 bit variables

Note: Remember that Bit variables are stored as part of a 16 bit word. There are 256 words.

The real-time status of Timers, Counters, Bits and Variables can be viewed and changed on the C and Ladder Variables form (Programming>Variables).

The unit will support up to 32K of compiled code. You may also store up to 32K of source code (optional) in the unit. However, note that compiled code is usually about half the size of the source code. Thus, if you have a ladder program which exceeds 16K compiled, you will most likely not be able to upload its source code.

Conserving Memory with Windows 95 and 98

From the main menu, select *File* and then *Conserve WinResources*. This feature conserves Windows resources to allow large programs (more than 500 instructions) to be used . Under Windows 95 and 98 it's recommended that you select this, especially if you have a large ladder program. Under Windows NT, deselecting *Conserve WinResources* will speed up program performance.

Starting a new Ladder Program

To run the Ladder Editor, click on *Ladder*, then *Open*, from the Control Center menu bar, or click on the *Ladder* button from the Control Center toolbar. Select a unit from the list, click either *OnLine* or *OffLine*, and then click *OK*.

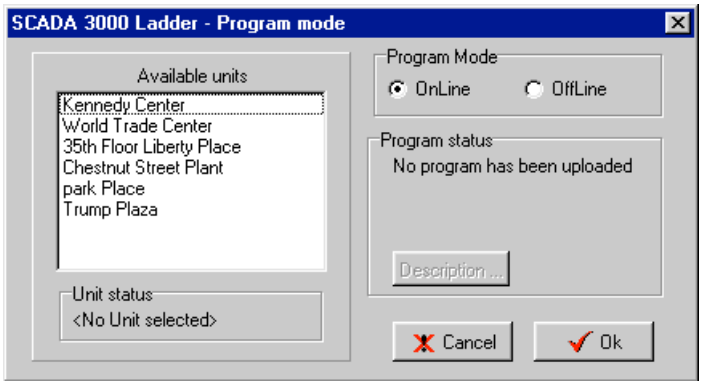


Figure 1: Ladder Program Mode

This will allow the program to intelligently allow you to select data points and check for invalid I/O addresses based on the programming in the selected unit. Click *File* from the main menu and select *New Program*. The following form will appear:

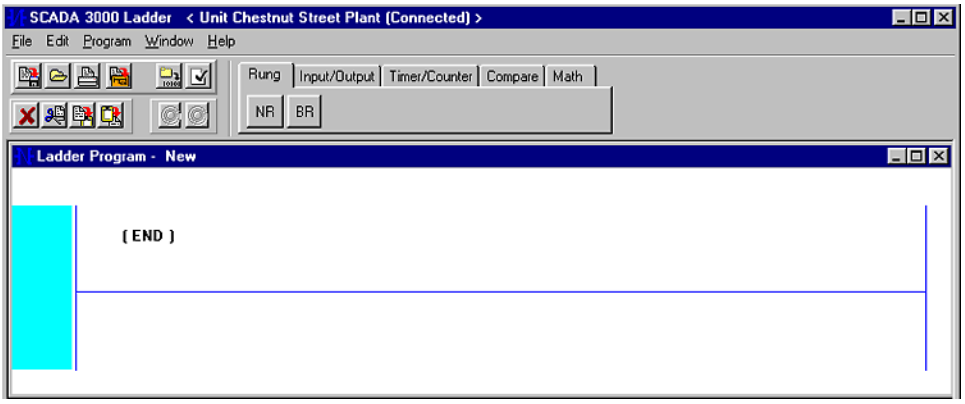


Figure 2: Ladder Editor

This is the Ladder editor screen from which you can build your ladder program. Several functions are available from this form including:

- Ladder program construction
- Ladder Program Save and load
- Ladder program compiling
- Ladder Program Upload and Download

- Ladder rules check
- Ladder Real-Time execution viewer

Adding Rungs

Once you open the Ladder editor you can begin editing a New program. A new program screen will only display an End rung. No symbols can appear on the end rung. To insert a new rung click and hold the left mouse button on the NR (New Rung) button from the rung tab and drag it to the yellow box on the left side of the window form. The box will turn green indicating that you can drop the New Rung in that location. You may add as many rungs as needed to your ladder program.

Instruction Toolbars

There are five tabbed categories which contain the toolbars for all of the ladder instructions. The Tabs are: Rung, Input/Output, Timer/Counter, Compare, and Math (See Figure 2). To place an instruction on a rung, choose the Tab containing it, then click the appropriate button and hold it as you drag it to the position where you want to place it on a rung. The yellow box will turn green indicating that you can drop the instruction (release the mouse button).

Continue building your program by adding more rungs and inserting instructions as required.

Branching

Branches allow you to create parallel instructions. You may have more than one branch per rung and even have multiple parallel branches. To place a branch, choose the Rung tab, click and hold the BRA button, and drag it to the rung where you want the branch placed. Drop the branch and move or insert the instructions within the branch as required. The yellow boxes will indicate where instructions or branches may be inserted. See the figure below:

Step 1. Add the branch to the rung.

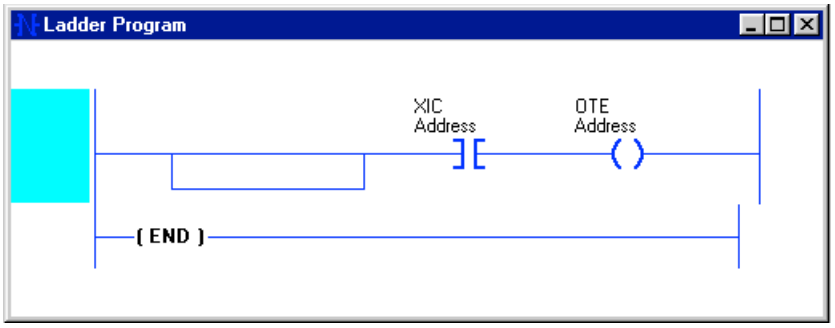


Figure 3: Adding a Branch

Step 2. Insert instructions on the branch.

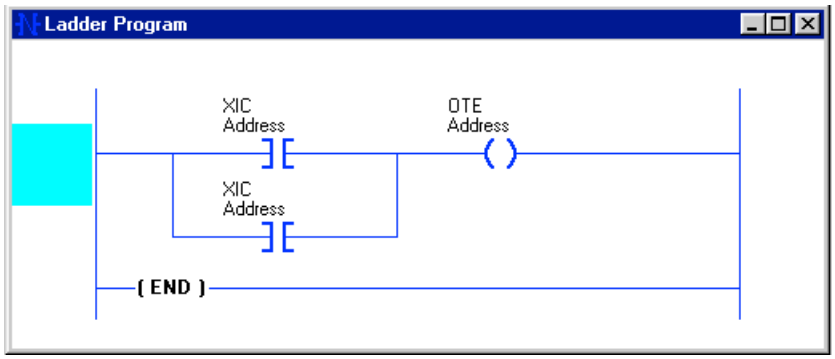


Figure 4: Adding Instructions

Entering Data Points, Titles, and Comments

Every instruction will have one or more Data Point addresses associated with it. Input and Output instructions will only have only one while math functions may have up to three. Title and comment text fields are generated automatically when you place an instruction. The title field is yellow, the comment field is green. Double-click on the title or comment field to edit or insert text. To select a data point or to edit an address, right-click on the instruction. A list of parameters will appear. Click on the parameter you wish to edit. A typical instruction sequence with UAF addresses and labels is shown below:

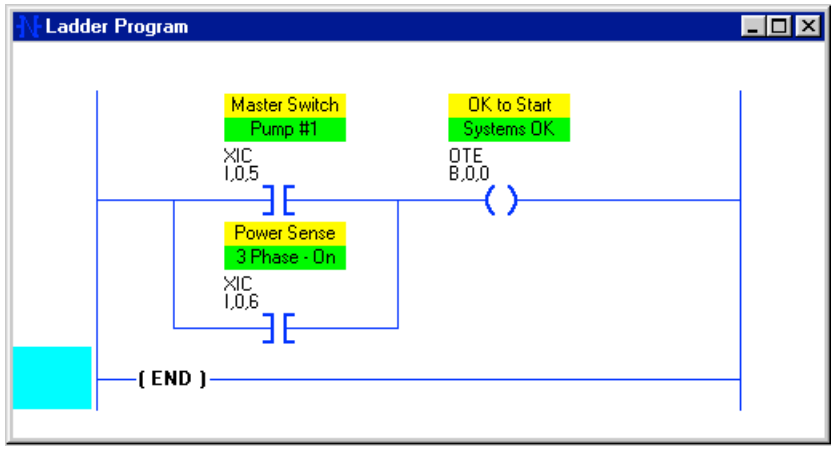


Figure 5: Adding Data Points & Titles

To add Rung Title and Comment fields, place the mouse at the intersection of the rung and ladder on the lefthand side. Right-click the mouse button and choose *Rung Title* or *Rung Comment*. Enter the text. Press *Enter*.

Note: Do not click *Delete* to remove a title or comment. *Delete* removes the entire Rung.

Changing Instructions

Instructions of the same type may be changed without re-entering addresses. For example, XIC can be replaced by XIO, CTU can be replaced by CTD, ADD can be replaced by SUB, and so on.

To perform an instruction change, click on a new Instruction button from the Toolbar. Then right-click on the instruction you wish to change and select Change from the menu that pops up.

Cut, Copy, Paste

The Ladder Editor supports *Cut*, *Copy*, *& Paste* operations for all ladder instructions. You may initiate these functions by right-clicking on the intruction or by selecting the instruction and then using the *Cut*, *Copy*, *Paste* buttons on the toolbar.

The *Cut* function will remove the instruction and store it so that you can *Paste* it elsewhere in your program. It will retain the addresses and presets associated with the instruction.

The *Copy* function will store a copy of the selected instruction so that you can *Paste* it elsewhere in your program. It will retain the addresses and presets associated with the instruction.

The *Paste* function will insert the instruction in memory at the location specified by the mouse. You must point the mouse to a valid hot-spot on a rung and then right-click. Hot-spots are located at the beginning of rungs, between instructions, and after the last instruction on a rung. A valid hot-spot will display a small black box. When the box is displayed, select *Paste* from the pop-up menu, or click on the *Paste* button from the toolbar.

Saving and Loading files

To save a Ladder Program click on *File* then *Save*, or click on the *Save* button from the toolbar. Enter a filename and click *OK*. The program with all of its titles, comments, and addresses will be saved.

To Load a previously saved ladder program, click *Program* then *Load* or click on the *Open* file button from the toolbar. Select the program file name you wish to load and click *OK*.

Deleting Rungs and Instructions

To delete rungs, right-click at the intersection to the left where the rung meets the ladder. Choose *Delete* from the menu; or highlight the rung by left-clicking at the intersection, and click the *Delete* button in the toolbar.

To delete instructions, click on the Instruction and click the *Delete* button from the toolbar; or right-click on the struction and choose *Delete* from the menu.

Syntax Checking

You can check your program for instruction and address errors by selecting the *Syntax Check* button from the toolbar, or by clicking *Program>Syntax Check* from the main menu. If any errors are found, the instruction with the error will be highlighted in red. The syntax checker will locate errors one at a time. Continue running the syntax check program until all errors are eliminated. Syntax check status will be displayed on the lower righthand side of the screen.

Compiling, Verifying and Uploading

The ladder program must be compiled before it can be uploaded to the SCADA 3000. The unit will support up to 32K of compiled code, with the option of also storing up to 32K of source code. Compiling translates the graphical instructions into a sequential file that can be read by the SCADA 3000. In addition, the program will be checked for missing or incomplete addresses and illegal logic. If any errors are detected they will be listed on the screen. Once you correct the errors be sure to re-compile the program before uploading it. To Compile the program click

on *Program*, then *Compile and Upload*, or click on the *Compile and Upload* button from the toolbar. Once your program compiles without errors, you can upload it by clicking the *Yes* button at the prompt. The software also allows you the option of uploading the source file into the SCADA 3000. The source file is the same file that can be saved on your computer, containing the graphical representation of your program. It also includes instruction addresses, titles and comments. Note that the compiled code is usually about half the size of the source code, so if your compiled code exceeds 16K, it's unlikely you will be able to upload the source code as it will likely be larger than the 32K limit.

To upload the source file with the compiled file, select *Program* from the main menu and click *Upload Source* so that a check appears next to it. If you upload the program now, both the compiled code and source code will be uploaded to the unit. This feature allows any user to download the complete ladder program.

Running and Stopping Ladder Execution

The ladder program is controlled by the Ladder and C-Program Start/Stop form located on the SCADA 3000 Main Programming form. This is accessed by clicking the Start/Stop button on the Main Programming toolbar, or by selecting *Programming*, then *Control* from the main menu.

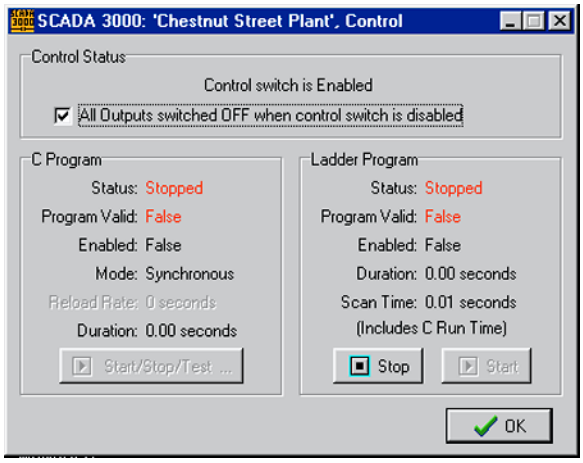


Figure 6: Start/Stop form

The Ladder and C-Program Start/Stop screen provides information, timing measurements and options. At the top of the screen is an option which determines how the outputs (relay & analog) are treated when the Control Switch is placed in the Disable position. If the option is not set, then the outputs will remain in the state they were in when the switch is placed in the Disable position. When the option is set, all unforced relay outputs will be switched OFF and all analog outputs will be set to their low table value. Forced outputs are unaffected by the position of the control switch. To enable this option, click in the option box at the top of the screen.

The Start/Stop form also indicates: if control programs are running or stopped; if valid Ladder or C-Programs are loaded; if control programs are enabled or disabled. It also provides control program timing information. The timing statistics are described below.

Duration: Amount of time for the Ladder or C-Program to execute.

Scan Time: Amount of time for the inputs to be read, plus the duration of the Ladder program, plus the duration of a synchronous C-Program, plus the time required to update the outputs.

Ladder Program Statistics

The statistics of the ladder program can be displayed after the program is successfully compiled or checked for syntax by enabling the *Display Usage* option in the program menu. This feature is useful to keep track of which Ladder Variables and Bits have been used. A summary of all instructions, addresses and instruction types is also provided. A sample report is shown below:

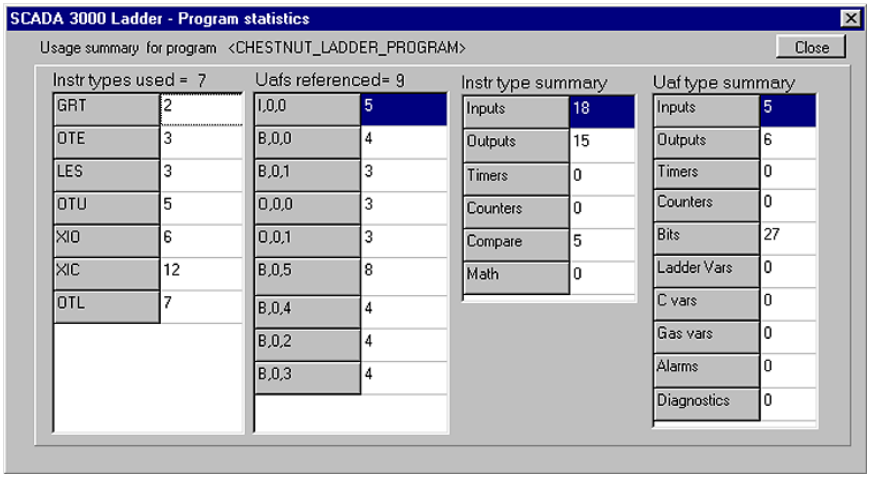


Figure 7: Ladder Statistics form

Ladder Run Viewer

The Ladder Run Viewer allows you to see your Ladder program execute on the screen in real time. In this mode, inputs and outputs will highlight as they become active, and all addresses representing numerical values will display those values. Status bits associated with timer and counter instructions will highlight as they become active.

To enable this mode, compile and upload your ladder program, then start it from the main unit programming form as discussed above. Next click the red *Run Program* button on the Ladder Editor screen, and then the green *Realtime Status* button. The symbols of your ladder program will now highlight when they are active. Math and comparison instructions will display the actual values. In this mode you cannot edit your program. To exit the realtime view mode, first click the green button and then the red button.

Printing

To print, select *File* in the Main Menu, and choose *Print*. Note that where the print dialog shows the number of pages, this actually indicates the number of rungs to be printed. You can print all or a part of your ladder program.

On regular, letter-size paper, the program prints 8 rungs per page.

Viewing and Forcing Ladder Variables, Counters, Timers & Bits

The real-time values for the Ladder Variables, Counters, Timers and Bits can be displayed using the Ladder & C Variables form. From the main menu click *Programming* then *Variables* or click on the *Ladder & C variables* button from the toolbar. Next select the Ladder Variables Tab. The following form will appear:

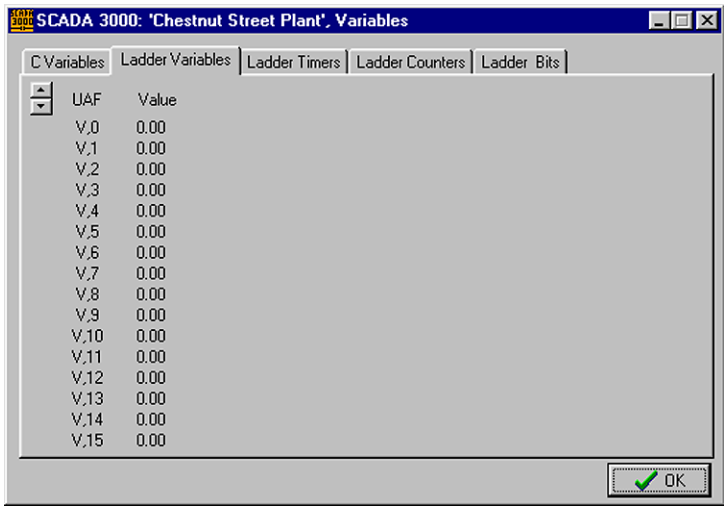


Figure 8: Ladder Variables form

Ladder Variables

The Ladder Logic program can use up to 128 floating point variables. This form will display the values of those variables. Use the Up/Down arrows to flip through the pages to access the variables you wish to view. To change the value of a variable, double-click on the value you wish to change. Enter the new value in the pop-up form and click *OK*.

Remote Variable Programming via Touch-Tone telephone

When you call into the SCADA 3000 and enter Voice Mode, a menu is spoken:

"To record message, press 1", "To play message, press 2", etc.

The menu option includes:

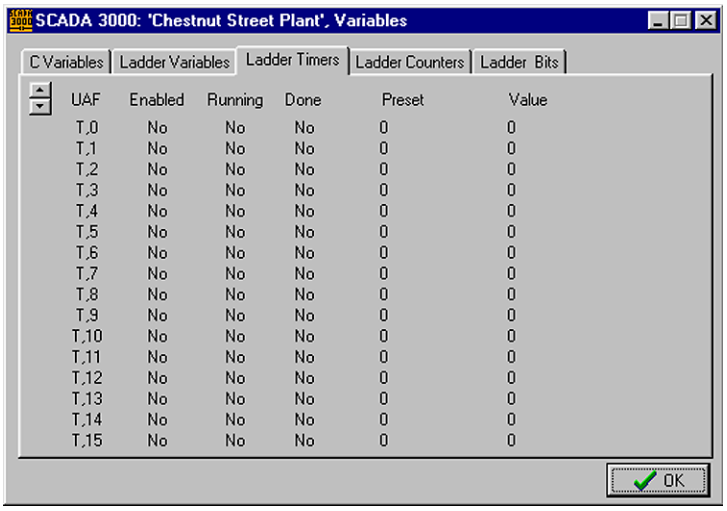
"To program ladder variable 127, press 9"

If you press '9' the unit will say "Variable 127 is now at ___, enter number followed by the pound key."

You can enter a single digit between 0 and 9 using your touch-tone telephone. The numerical value you enter will be copied into ladder variable 127. This allows you to remotely initiate a control sequence within your ladder program simply by adding statements that check the value of ladder variable 127. Once the desired logic has executed, you can reset ladder variable 127.

Ladder Timers

The Ladder Logic environment has 64 timers which can be used as necessary in your Ladder program. The Ladder Timers form will display the parameter values for each timer, 16 at a time. Use the Up/Down arrows to flip through the pages to access the Timer values you wish to view. To change the value of a timer, double-click on the value you wish to change. Enter the new value in the pop-up form and click *OK*.

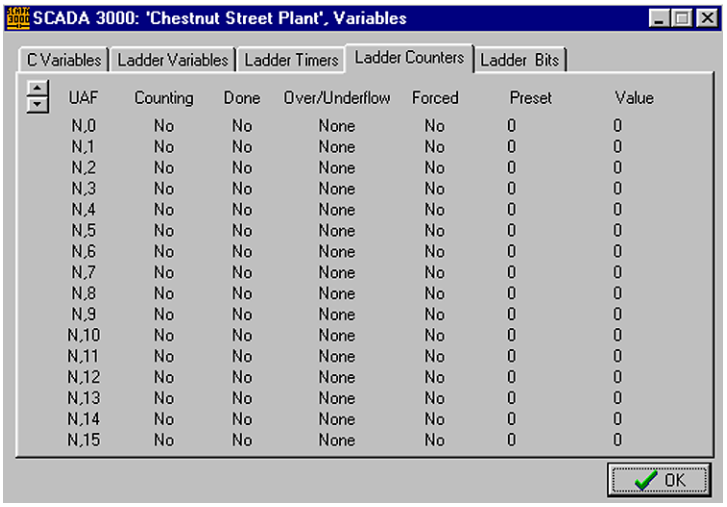


UAF	Enabled	Running	Done	Preset	Value
T,0	No	No	No	0	0
T,1	No	No	No	0	0
T,2	No	No	No	0	0
T,3	No	No	No	0	0
T,4	No	No	No	0	0
T,5	No	No	No	0	0
T,6	No	No	No	0	0
T,7	No	No	No	0	0
T,8	No	No	No	0	0
T,9	No	No	No	0	0
T,10	No	No	No	0	0
T,11	No	No	No	0	0
T,12	No	No	No	0	0
T,13	No	No	No	0	0
T,14	No	No	No	0	0
T,15	No	No	No	0	0

Figure 9: Ladder Timers form

Ladder Counters

The Ladder Logic environment has 64 counters which can be used as necessary in your Ladder program. The Ladder Counters form will display the parameter values for each counter, 16 at a time. Use the Up/Down arrows to flip through the pages to access the Counter values you wish to view. To change the value of a counter, double-click on the value you wish to change. Enter the new value in the pop-up form and click *OK*.



UAF	Counting	Done	Over/Underflow	Forced	Preset	Value
N,0	No	No	None	No	0	0
N,1	No	No	None	No	0	0
N,2	No	No	None	No	0	0
N,3	No	No	None	No	0	0
N,4	No	No	None	No	0	0
N,5	No	No	None	No	0	0
N,6	No	No	None	No	0	0
N,7	No	No	None	No	0	0
N,8	No	No	None	No	0	0
N,9	No	No	None	No	0	0
N,10	No	No	None	No	0	0
N,11	No	No	None	No	0	0
N,12	No	No	None	No	0	0
N,13	No	No	None	No	0	0
N,14	No	No	None	No	0	0
N,15	No	No	None	No	0	0

Figure 10: Ladder Counters form

Ladder Bits

The Ladder Logic environment has 4096 bit variables which can be used as necessary in your Ladder program. Remember that Bit variables are stored as part of a 16 bit word. There are 256 words. The Ladder Bits form will display the 16 bit values for each word, 16 words to a page. Use the Up/Down arrows to flip through the pages to access the Bit values you wish to view. To change the value of a bit, double-click on the value you wish to change. Enter the new value in the pop-up form and click *OK*.

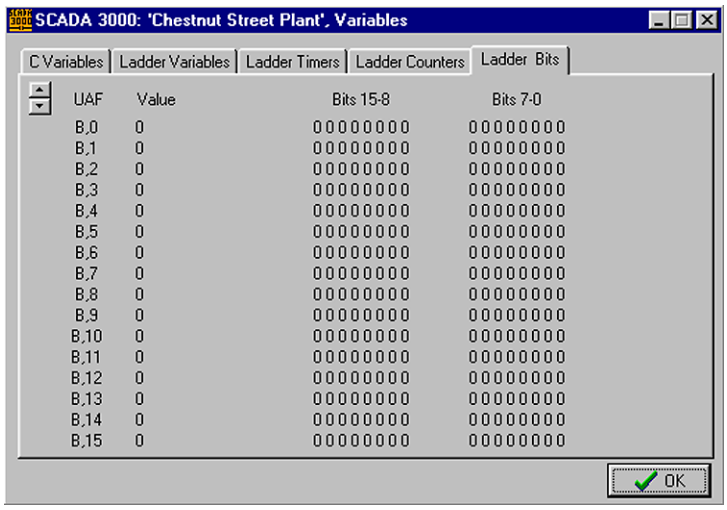


Figure 11: Ladder Bits form

Ladder Program Performance

The execution time of the Ladder Program will depend on several factors. For example, simple on/off instructions will run faster than mathematical functions. The sequence of events when the ladder program runs goes like this: (1) Scan all inputs, (2) Run ladder Program, (3) Update outputs. Listed below are some typical time requirements for estimating the total run time of a ladder program.

Input Scan

Main Board (all 16)	5.0ms (digital), 12.5ms (analog)
Universal Input Module	3.5ms per module
Pulse Count Module	3.2ms per module
Thermocouple Module	3.9ms per module

Run Ladder Program

100 instruction program	2.2ms per 100 instructions

Update Outputs

Main Board Relays (all 8)	1.0ms
Relay Output Module	0.9ms per module
Analog Output Module	2.4ms per module

Examples

Program A (small)	Total Run Time
Main Board with no modules All inputs are digital 100 instruction ladder program.	8.2ms or 122 scans per second

Program B (medium)	Total Run Time
Main board using all analog inputs plus a Universal Input Module and an Analog Output Module plus a 200 instruction ladder program	23.8ms or 42 scans per second

Program C (large)	Total Run Time
Main board using all analog inputs with 8 Universal Input Modules, 4 Relay Output Modules, 2 Pulse Count Modules, 1 Analog Output Module, and a 1000 instruction ladder program	75.9ms or 13 scans per second

Ladder Program Examples

Example #1

The ladder program below (Figure 5) illustrates simple input to output control. If input 0 closes, then output 0 will turn on. If Input 0 opens, then the output will turn off.

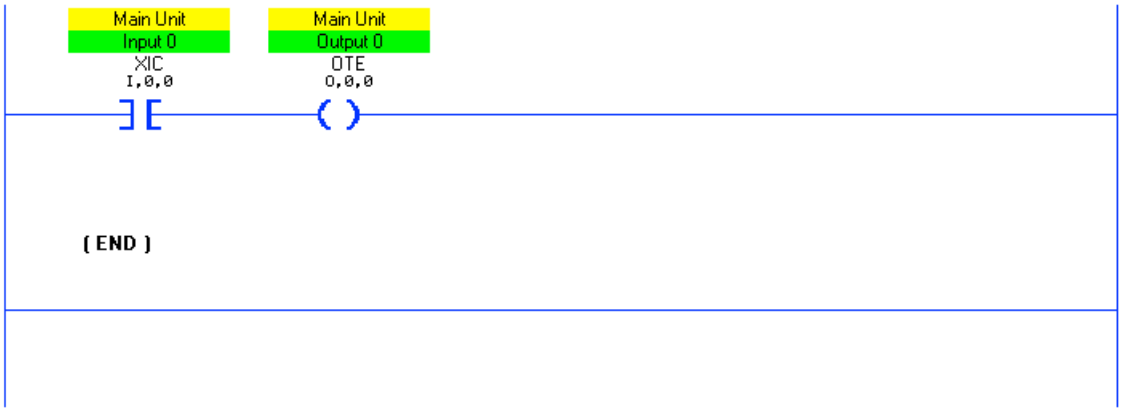


Figure 12: Simple input to output control

Example #2

The following program (Figure 6) would be used to refill a tank when it becomes empty. Say the tank has a float switch at the top to indicate that the tank is full, and a float switch at the bottom to indicate that the tank is empty. When the level gets too low, the low-level float switch opens, making the XIO instruction true, which latches the pump on. The pump will stay on until the high-level switch closes, causing the XIC instruction to become true, which unlatches the pump and turns it off.

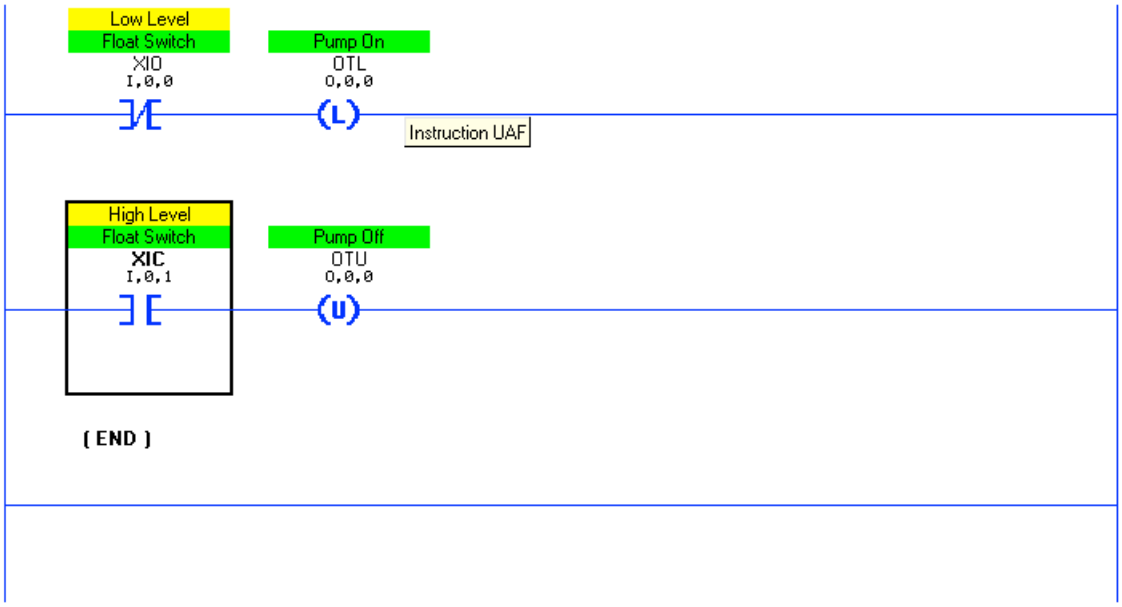


Figure 13: Float Switch control

Example #3

In this example, a timer is used to delay the start of a system by 10 seconds. When the system power switch is turned on, the timer starts running. When it reaches the preset value, the timer-done bit is set, which makes the XIC instruction on rung #2 true. This in turn activates output 2, which turns on the system.

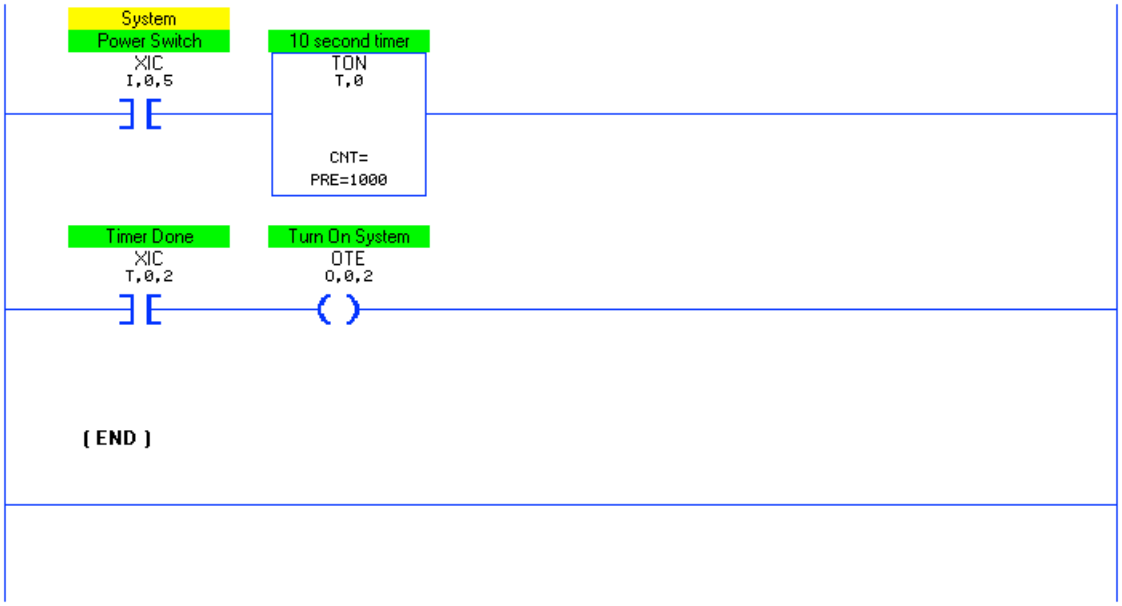


Figure 14: Timer Delay

Example #4

The following program uses a Count-Up and a Count-Down instruction to keep track of the number of acceptable widgets produced. As a widget comes down the production line, an optical sensor on input 6 senses the widget and makes the XIC instruction true, which increments Counter 5. When widgets are rejected, they are diverted down another conveyor with an optical sensor on input 10. When the reject sensor sees a widget, it decrements Counter 5. With this configuration the counter is able to keep track of the acceptable widgets produced. A reset button on input 12 resets the counter to 0.

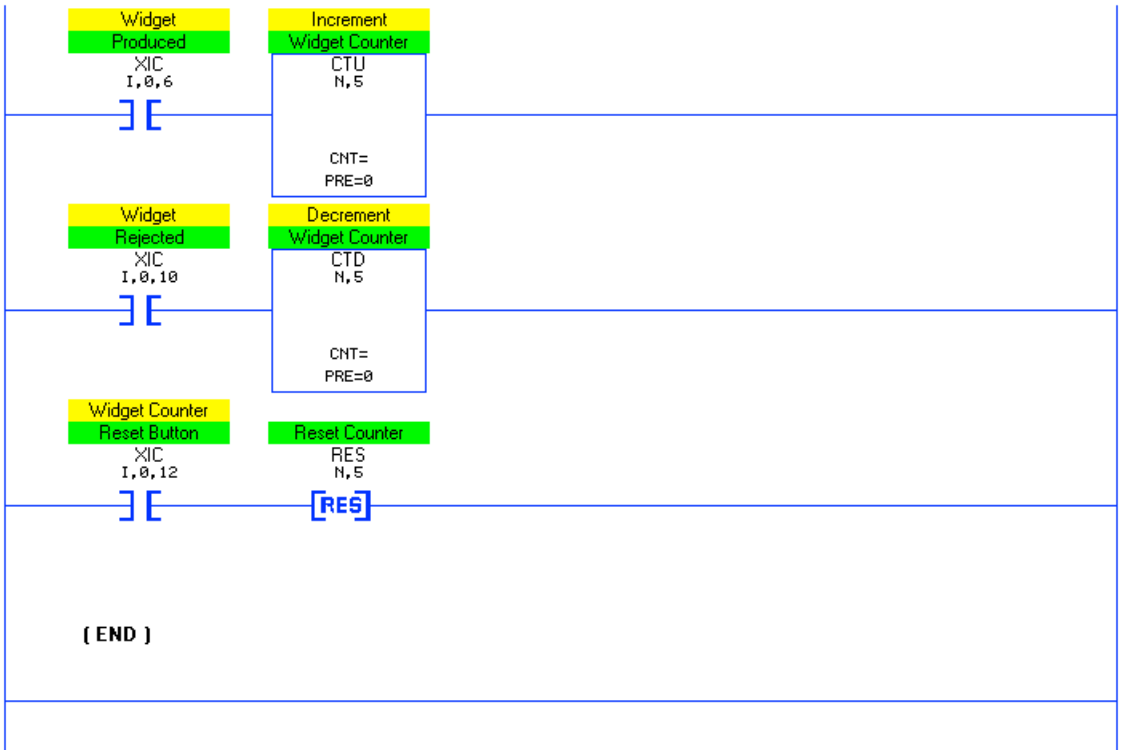


Figure 15: Count-Up/Count-Down instruction

Example #5

The following program will monitor a normally-open push-button switch connected to input 10. When the switch is depressed the program will reset (acknowledge) alarm 0. The OSR instruction prevents the reset instruction from being executed continuously and allows the alarm to proceed if the switch were to become shorted.

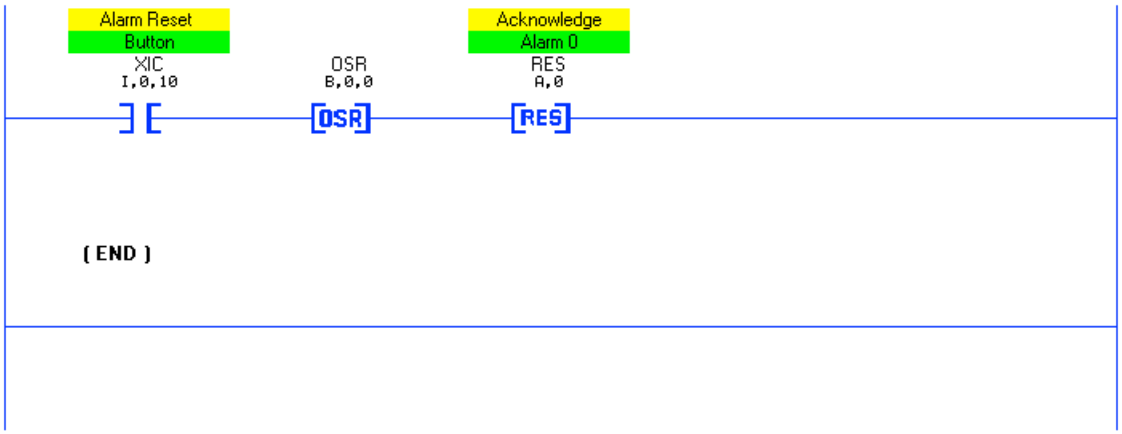


Figure 16: OSR instruction

Chapter 16: PROGRAMMING IN C

C is a computer programming language that has become increasingly popular in industrial automation for writing sophisticated control programs. Integrated within SCADA 3000 is a C language interpreter. This interpreter allows you to write and compile a program written in the C language to perform intelligent input/output control and mathematical functions. Note that the C interpreter in the SCADA 3000 is a slimmed-down version of C and that all functions normally associated with ANSI C will **not** be available. Listed within this chapter are explanations on functions that are available along with examples on how to make use of them.

Specifications

The maximum size of the C program is 32K of text and/or 16K of compiled text. Up to 1024 variables may be defined and each variable name can be up to 15 characters. The SCADA 3000 also has six predefined variables for date and time reference. The C program can be set to run synchronously or asynchronously with the ladder program. In synchronous mode the C-program will run continuously or, if the ladder program is running, synchronized with the ladder program. In asynchronous mode the C program will run on an independent time schedule.

ABOUT THE C LANGUAGE

C is a structured programming language that is similar to other structured languages like Pascal. The **structure** of C is like asking a question or posing a problem, and then going through the simple steps of information gathering and action to solve the problem. For example, if your alarm clock is ringing and you want it to stop ringing, you must hit the OFF button to turn it off, otherwise nothing will happen. If you want the SCADA 3000 to turn off the alarm clock, you must first tell it how to recognize that it is ringing, then give it the command to turn it off when it recognizes that it is ringing, otherwise do nothing.

To give SCADA 3000 this information, you must translate English commands into C commands. SCADA 3000's built-in compiler then translates the C commands into SCADA 3000's language.

The C language structure is fairly simple. Each program must have a beginning and an end. Each individual statement within the program must open, provide its information, and then close. Think of it as writing sentences—without the proper sentence structure and punctuation, you will get syntax errors. The reader (SCADA 3000) will not be able to receive the message or understand what to do.

The SCADA 3000 C interpreter comprises the following elements:

STRUCTURE: To write a valid C program, regardless of length, you must follow its simple structural elements.
KEYWORDS: These are short words or symbols that define variables or execute specific actions within a C program. They include operators, assignment, comparators, and comments.
PREDEFINED VARIABLES: These are variables that have specific predefined values that are automatically updated and cannot be changed by the user.
FUNCTIONS: These are word commands that perform a predefined function within a C program.

ARRAYS: This is a more advanced programming type that significantly shortens and simplifies long programs.
ERROR HANDLING: When a program is compiled, it is scanned for language related errors. An error message, including the type of error, is displayed.
EDITING TOOL: The SCADA 3000 Windows Software provides an editing utility that allows you to write, compile, upload, download, run, and start C programs.

This chapter provides basic instruction in the C language and gives all the commands that are valid for use with SCADA 3000. Sample programs are included. If you have questions or problems, please contact our technical support department at (610)558-2700 or via email at support@sensaphone.com.

STRUCTURE

Listed below are some examples that will help demonstrate the C language structure. Refer to the following pages for explanation of the keywords, functions, and commands used within the sample programs.

1) All programs must begin with the main () function. It must be followed by an open brace { and closed with an end brace }.

```
Main ()
     {
     }
```

2) A statement is a line of programming code. All statements are placed inside the braces {} and end with a semicolon (;). For example:

```
main ()
    {
    write_uaf(output,0,0,on);
    }
```

This program will turn on output relay 0, on the main board.

3) In the C language, numbers are stored in what are called variables. A variable must be defined before the main () function. SCADA 3000 allows you to have up to 1024 variables.

```
intx;
   main ()
         {
         x=12
         }
```

In this program we defined a variable called "x" and set its value equal to 12. Here, x is defined as an integer. SCADA 3000's language can define variables as characters (char), integers (int), or floating point numbers (float). A character can hold a value from -128 to 127 and an integer can hold a value from -32,768 to 32,767. Both characters and integers must be whole numbers. A floating point number is one that can have a fractional part (i.e., numbers after the decimal point). Floating point numbers can be very large or very small, generally between 10^{-38} and 10^{+38}. Generally speaking, there is no advantage to using variables as *int* or *char* types within the SCADA 3000, and examples from this point on will use only variables of the type *float*.

The following program will set x equal to the value of input 0 on the main board.

```
float x;
   main ()
         {
         x=read_uaf(input, 0,0);
         }
```

In many of the examples in the chapter we use x and y as variable names. The variable names can be up to 15 characters long. Numbers can also be used as long as the name does not begin with a number. In addition to letters and numbers, the underscore character may also be used in a variable name, but it cannot be the first character of the variable name. You cannot have spaces in the middle of a variable name.

The following are valid variable names:

```
float average;
   float outside_temp;
   float condition1;
   float contact12;
```

The following are *not* valid variable names:

```
   float 3temp;
   float average_result_per_day;
   float inside room;
   float _timer;
```

4) The SCADA 3000 C language has other tools available to construct your C program. One of the most important tools is the "if" statement. The "if" statement is used to make a decision whether or not to execute a sequence of statements.

Example: The following program will turn on output 0 only if x is greater than 100, which in this case is true. When the condition in the "if" statement is true, the program continues through the statements inside the "if" braces. If the condition was not true, the program skips the block of statements.

```
float x;
main ()
   {
   x=120;
   if (x>100)
      {
      write_uaf(output,0,0,on);
      }
   }
```

The "else" statement can be used with the "if" statement and gives instructions on what to do when the "if" statement is false.

Example: In the following program, if x is greater than 100, then output 0 on the main board will be turned on. If x is not greater than 100, then output 0 on the main board will be turned off. In either case, output 3 on the main board will always be turned on.

```
float x;
main ()
   {
   x=75
   if (x>100)
      {
      write_uaf(output,0,0,on);
      }
   else
```

```
    {
    write_uaf(output,0,0,off);
    }
write_uaf(output,0,3,on);
}
```

There are C commands other than "if" and "else", and built-in variables, to access SCADA 3000's inputs and outputs. Details for these will be covered later.

KEYWORDS

The following is a list of all valid components of the C language within the SCADA 3000.

```
char
int
float
if
else
for
do
while
```

Definitions

CHAR - Used to define a variable as a character. A character can hold a value from -128 to +127 and must be a whole number. Exceeding this range will cause incorrect results. Note: Variables defined as type CHAR will not save memory space within the SCADA 3000.

Example:

```
char x;
main ()
   {
   x=12;
   }
```

INT - Used to define a variable as an integer. An integer can hold a value from -32,768 to +32,767 and must be a whole number. Exceeding this range will cause incorrect results. Note: Variables defined as type INT will not save memory space within the SCADA 3000.

Example:

```
int result;
main ()
   {
   result=17;
   }
```

FLOAT - Used to define a variable as a floating-point number. A floating-point number is one that can have a fractional part (i.e., numbers after the decimal point). Floating-point numbers can also be very large or very small, generally between 10^{-38} and 10^{+38}. Numbers of this size can be represented using exponential notation. In exponential notation, the significant digits of a number are raised to a power of ten. The basic form comprises a signed series of digits including a decimal point, followed by the letter **E**, and then by a signed exponent to indicate the power of ten being used.

Examples:

```
1,987          =     1.987E3
120,000,000 =     1.2E8
.000000314   =     3.14E-7
-0.00000000005 =  -5.0E-11
```

IF - Used to make decisions.

Example:

```
main ()
   {
   if (read_uaf(input,0,0)>100)
      {
      write_uaf(output,0,0,on);
      }
   }
```

ELSE - Used with IF to execute a statement when the IF condition is false.

Example:

```
main ()
   {
   if (read_uaf(input,0,0)>100)
      {
      write_uaf(output,0,0,on);
      }
   else
      {
      write_uaf(output,0,0,off);
      }
   }
```

FOR - Used to execute a statement (or statements) multiple times. Contains a start condition, a stop condition, and a control statement. The following example starts a counter at one, checks that it is less than nine, and executes the output statement. Then it adds one to the counter and checks that it is still less than nine. When the counter equals nine, the FOR loop is finished. The output statement in this program turns outputs 0 through 8 off, on the main board.

```
float count;
main ()
   {
   for (count=0; count<9; count=count+1)
      {
      write_uaf(output,0,count,off);
      }
   }
```

DO - Used to execute a list of statements while a condition is true. The statements are always executed at least once. The following example always sets output 0 on and keeps it on as long as input 0 is greater than 100.

Example:

```
main ()
   {
```

```
do {
     write_uaf(output,0,0,on);
     } while (read_uaf(input,0,0)>100);
}
```

WHILE - Used to execute a list of statements while a condition is true. The statements are only executed if the condition is already true. The following example only sets output 0 on when input 0 is greater than 100, and keeps it on until input 0 is less than or equal to 100.

Example:

```
main ()
   {
   while (read_uaf(input,0,0)>100);
      {
      write_uaf(output,0,0,on);
      }
   }
```

OPERATORS - Symbols used to execute mathematical operations and determine whether a particular condition exists.

```
+     Adds two values
-     Subtracts two values
*     Multiplies two values
/     Divides two values
%     Finds the remainder of a division
||    Checks if one condition or another exists
&&    Checks if one condition and another exists
```

ASSIGNMENT - Symbol used to assign a numeric value.

```
=   assigns a new value to a variable
```

COMPARATORS - Symbols used to compare numeric values to each other

```
<    Less than
>    Greater than
==   Is equal to
!=   Is not equal to
<=   Less than or equal to
>=   Greater than or equal to
```

COMMENTS - Text used to add notes to a program. Typically used to briefly explain what the program or statement is doing. They may be used freely to make a program easier to understand. Comments are ignored by the compiler.

Note: Do not use the characters "*" or "/" within the body of a comment.

```
/*   Begin comment
*/   End comment
```

PREDEFINED VARIABLES

SCADA 3000's C language has several variables that are predefined. These variables are automatically updated with the proper information outside of the C program and cannot be changed by the user. They include:

input	year
output	month
timer	day
counter	day_of_week
bit	hours
ladder_variable	minutes
c_variable	seconds
gas_flow	on
alarm	off
diagnostic	

MONTH, DAY, YEAR, HOURS, MINUTES, & SECONDS

These variables represent the values from the real-time clock. They are defined as integer type. Their internal definitions look like the following:

```
int year;
int month;
int day;
int day_of_week;
int hours;
int minutes;
int seconds;
```

Note: You do not need to define these at the beginning of your program to use them.

Description:

YEAR - Holds the present value from 1999 to 2100.
MONTH - Holds the present value from 1 to 12.
DAY - Holds the present value from 1 to 31.
DAY OF WEEK-Holds the present value from 0(Monday) to 6(Sunday).
HOURS - Holds the present value in 24 hour time from 0 to 23.
MINUTES & SECONDS - Holds the present value from 0 to 59.

Example: This program will turn on output 0 at midnight.

```
main ()
   {
   if ((hours==23)&&(minutes==59))
      {
      write_uaf(output,0,0,on);
      }
   }
```

FUNCTION LIBRARY

In SCADA 3000's C language there are five functions defined - Read, Write, Reset, Power, and Write Record. These functions allow you to retrieve/assign certain values and incorporate them into your C program. Most parameters within the SCADA 3000 are accessible via the UAF (Universal Address Format). The Read and Write functions always include an associated UAF. A function can be a statement by itself or it can be used to retrieve a value and return it.

READ FUNCTION - Used to read a value from the SCADA 3000 as described by its UAF.

Summary:

```
float read_uaf(t,g,e)
int t;  UAF source type: input, output, timer, counter, bit,
        ladder_variable,gas_flow,alarm,diagnostic
int g;  UAF group number:        0-1023
int e;  UAF element number:      0-255
```

Description

The read_uaf function returns the value of any UAF register that is specified by the three parameters, The value of the specified UAF will always be returned by this function as a float even if the real UAF register is an integer or bit. Note that there is no run time error checking on this function. If a user requests an invalid register location, the function will not generate an error, but will just return an invalid result.

RETURN VALUE - The read_uaf function returns the value of the specified UAF register.

Example:

```
float tank_high;
main ()
   {
   if (read_uaf(input,0,0)>100.0)      /*If input 0 on main
board is greater than 100   */
      {
      tank_high=1;                    /* then set the tank_high
variable equal to one   */
      }
   }
```

WRITE FUNCTION - Used to write a value to the SCADA 3000 as described by its UAF.

Summary:

```
float write_uaf(t,g,e,d)
int t; UAF source type:  output, bit, ladder_variable,
       c_variable, alarm
int g; UAF group number:   0-1023
int e; UAF element number: 0-255
int d; New data for UAF:   "on," "off," floating point number
```

Description

The write_uaf function is for changing the value in any of the UAF registers. The parameters of the function specify the location of the UAF register to be changed as well as the new value for the register.

RETURN VALUE - The write_uaf function always returns a zero.

Example:

```
main ()
   {
   if (read_uaf(input,0,0)>100.0)      /*If input 0 on main
board is greater than 100   */
      {
      write_uaf(output,0,0,on);            /* then set relay
output 0 on the main board to on   */
      }
   }
```

RESET FUNCTION - Used to reset (acknowledge) the specified alarm in the SCADA 3000.

Summary:

```
int reset(x)
int x; alarm number: 0-63
```

Description

The reset function is used for acknowledging any alarm in the unit. The parameter of the function specifies the alarm number to be acknowledged.

RETURN VALUE - The reset function always returns a zero.

Example:

```
main ()
   {
   if (read_uaf(input,0,0)==1)      /*If input 0 on main board
is closed  */
      {
      reset (0);         /* then acknowledge alarm 0  */
      }
   }
```

POWER FUNCTION - Mathematical function used to raise a value to the specified power.

Summary:

```
float pow (x,y);
float x:
float y;
```

Description

The Power function is used to raise a value X to the power specified by Y. The X parameter is the operand and the Y parameter is the exponent.

RETURN VALUE - The Power function returns x raised to the y power.

Example: The following program will take the square root (1/2 power) of the value of input 3 on the main board and store it in the variable c.

```
float a;
float b;
float c;
main()
   {
   a=read_uaf(input, 0.3);
   b=1/2;
   c=pow(a,b);
   }
```

WRITE_REC FUNCTION - used to write a single DataLog record to the DataLogger.

Summary:

```
write_rec();
```

Description

The write_rec function is used to write a single DataLog record to the DataLogger. All values selected for logging on the DataLogger programming form will be logged once when this function executes, regardless of whether or not the DataLogger is actually running or not. This

function is useful for event driven datalogging functions such as when an I/O point goes out of range or when an alarm exists. Do not allow this function to execute continuously or large amounts of data will accumulate in the datalogger. Also, do not allowe this function to execute more often than once a second because the DataLogger can only resulve timestamps to the nearest second and will throw out records with identical time stamps.

Example:

```
/* This program will write a datalog record once per minute */
/* if input 0 is greater than 100 */
int last_minute;
main ()
{
if (read_uaf(input,0,0)>100.0)      /* if input 0 is greater than 100*/
   {
   if (minutes != last_minute)      /* is it a new minute? */
         {
         write_rec();               /* then write a datalog record */
         last_minute=minutes;       /* reset minute check */
         }
   }
}                                   /* End of program */
```

ARRAYS

Arrays allow you to store a lot of related information in a convenient, organized fashion. An array lets you use one line of programming to create a series of variables. These variables share the same basic name and are distinguished from one another by a numerical tag.

Example:

```
float count[10];
```

This means that an array named count has 10 members or "elements", with each element having its own value, starting with 0 and ending with 9. The first element is count[0], the second element is count[1], and so on up to count[9]. The type float means that the actual numerical value of each element is a float.

Example:

```
count[2] = 2.45

count[4] = 55000

count[9] = -200000
```

SAMPLE PROGRAM

This program calculates a one-hour average temperature. The array named "numbers" sets up a series of variables from 0 to 60 to hold a value for input 0 for each minute in an hour. The 60 values are totaled, then averaged. The average can then viewed from the C & Ladder Variables form, stored in the datalogger, or displayed on the Real-time screen.

By using an array, the program becomes substantially more concise. The program is first listed, then a section by section explanation on how the program works is provided.

```
float numbers[60];   /* array: input 0 value for each minute */
int x;               /* index to the array */
```

```
float total;         /* total of the input 0 values */
float average;       /* total/60 */
   int oldminute;          /* minute counter */
   main()
      {
      if (oldminute != minutes)
         {
         oldminute=minutes;
         numbers[minutes] = read_uaf(input,0,0);
         total = 0;
         for (x=0;x<60;x=x+1)
            {
            total = total + numbers[x];
            }
         average = total/60;
         }
      }
```

The following is a step by step explanation of the program.

1)
```
if (oldminute != minutes)
    {
    oldminute=minutes;
```

This checks the value of minutes to see if a minute has passed. If a minute has passed, the value of oldminute is reset to the value of the new minute.

2) `numbers[minutes] = read_uaf(input,0,0);`

This line sets the value of the variable "numbers" for a particular minute to the current value of input 0. For example,

```
It is 3 minutes into the hour.

The value of input 0 is 72.
```

Therefore,

numbers[minutes] = read_uaf(input,0,0);

is equal to

numbers[3] = 72

3) `total = 0;`

This line initializes the variable "total" to zero.

4) `for (x=0;x<60;x=x+1)`

This line initializes the index variable (x) to zero, checks to see if it is less than 60, runs the next line, and then increments x by one. This line creates a loop that runs the following line until x reaches 60.

5) `total = total + numbers[x];`

This line of the program is very important. It eliminates having sixty lines of programming to

get a total of all the minute readings. The x serves to automatically read in the next value for each element in the array. For example:

```
The value of x is 6 (meaning it is minute 6, incremented in step
4).

The value of input 0 for minute 6 will be read into the array and
added to the total of the other 5 minutes.

   The value of input 0 at minute 6 is 70.

Therfore: numbers[6] = 70

   If the previous total was, say 351, then:     total = 351 + 70

   The new total is 421:   total = 421
```

6) When x reaches 60, the FOR loop is finished and the average is calculated.

```
average = total/60;
```

ERROR HANDLING

When a program is compiled it is scanned for language related errors. When an error is encountered, the ERROR is displayed followed by the line number. The line number may not be the exact line that contains the error. Sometimes an error is detected a line or two after the actual mistake. If an error occurs while compiling, the compiler aborts. It is possible to pass the compiler with no errors, but receive an error when the program is running. A run time error occurs when the syntax of a built-in function is incorrect. When a run time error occurs, the program will not run correctly.

Listed below are the possible compiling errors:

```
Description
Syntax
Semicolon expected
Unbalanced braces
Parentheses expected
While expected
Quote expected
Variable not found
Too many variables defined (limit is 1024)
Bracket expected
Compiled file too big
Reserved word declared as variable
```

C EDITOR

The SCADA 3000 Software contains a C programming utility that allows you to write C programs, upload/download them to the SCADA 3000, save/load C programs on your computer, compile, run, and start programs in the SCADA 3000. This can be performed while online through the RS232 serial port or remotely via modem. To access the C Editor, open the Control Center form and click on the "C" button inthe toolbar or choose "C" from the main menu. The following form will appear:

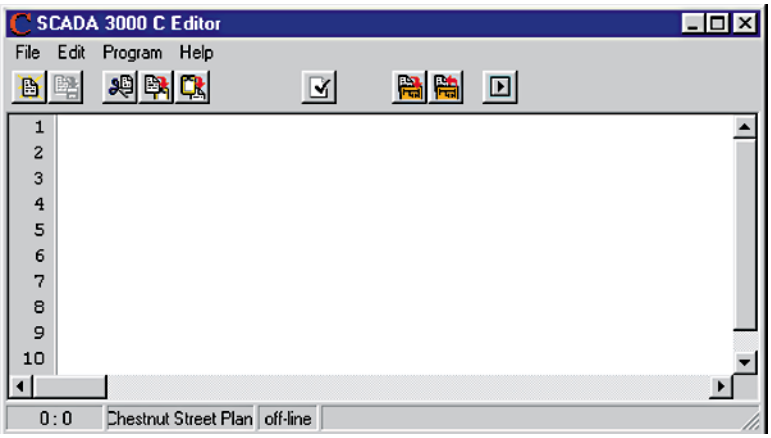


Figure 1: C Editor form

Use this form to write and edit your C programs. You do not have to be on-line with the SCADA 3000 to write a C program. The buttons on the form are, from left to right: Open Program, Save Program, Cut, Copy, Paste, Undo, Syntax Check, Compile Program, Upload Program, Download Program, and Start/Stop Program.

An example of a C Program written in the editor window follows.

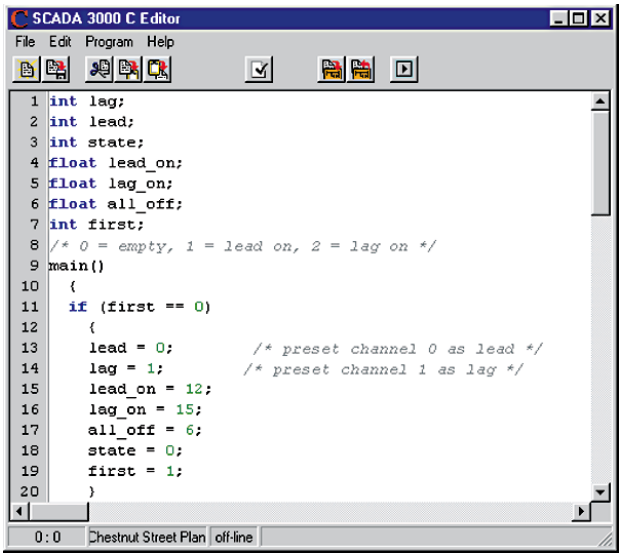

SCADA 3000 C Editor
File Edit Program Help
1 int lag;
2 int lead;
3 int state;
4 float lead_on;
5 float lag_on;
6 float all_off;
7 int first;
8 /* 0 = empty, 1 = lead on, 2 = lag on */
9 main()
10 {
11 if (first == 0)
12 {
13 lead = 0; /* preset channel 0 as lead */
14 lag = 1; /* preset channel 1 as lag */
15 lead_on = 12;
16 lag_on = 15;
17 all_off = 6;
18 state = 0;
19 first = 1;
20 }
0 : 0
Chestnut Street Plan
off-line

Figure 2: C program Example

When using the read_uaf and write_uaf functions, there is a shortcut to help construct these statements. From the C Editor menu select *Program*, then *Select Unit*. Choose the unit that will be runnning this C Program from the drop-down box, click the button that says *Log-in Now*, and click *OK*. When you are ready to enter your UAF, click on your right mouse button. Click on the *Insert UAF* parameter and a list of all valid UAFs will be displayed. Click on the UAF you want to use and it will be inserted automatically at the cursor point.

When you are finished writing your program, save it to your computer by clicking *File*, then *Save*. To open a previously saved program, use the *Open Program* button or click *File>Open* .

Uploading, Compiling, & Starting

Once you have finished writing your C program you can upload it to the SCADA 3000. To upload your C program go on-line with the SCADA 3000. Once on-line, open the C editor as

shown above. Open the program you wish to upload to the SCADA 3000. Click the *Upload* button from the toolbar or select *Program*>*Upload* from the main menu. Your C program will start transferring to the SCADA 3000.

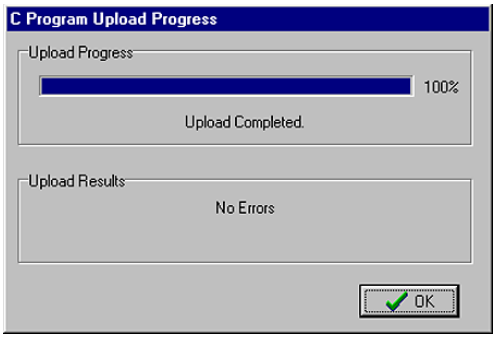


Figure 3:C Program Upload form

If your SCADA 3000 is on a radio network, an additional screen will be displayed so that you can select the SCADA 3000 that you want the C program sent to. Once the transfer is complete SCADA 3000 will automatically compile the program. This procedure is required to translate the program into a language that your SCADA 3000 can understand. When the program is compiled, SCADA 3000 scans your program, looks for errors, and converts it to a new format. If a problem is found, an error code will be displayed. The type of error code depends on where the error is found and what kind it is. Multiple errors must be corrected *one at a time* before SCADA 3000 will continue to scan the program. Please refer to the ERROR HANDLING section to identify the types of errors in your program. Each time you make a change to your C program you must Upload/Compile the new version.

NOTE: All C variables are reset to zero when the program is compiled.

Downloading

The C-Program source code can be downloaded from your SCADA unit by clicking the download button from the C-Program toolbar, or by selecting *Program*, then *Download* from the C-Editor main window.

Run

After the program has been compiled, the following form will appear:

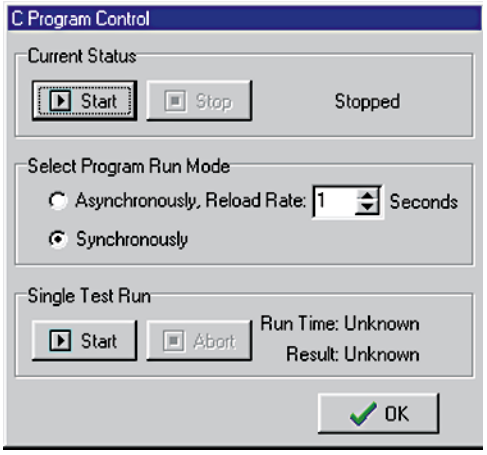


Figure 4: Start C Program form

Click on *Run Once* to test-run your program one time. The form will display The Run Time--the amount of time it took for your program to complete, and whether or not it ran from beginning to end OK. If not, it will display any errors encountered. The run time is useful in deciding whether to run the C program in synchronous mode or asynchronous mode. It will also help you to decide what to enter for the reload time if you choose asynchronous mode.

Start & Stop

After the program has been compiled, run once, and all errors have been corrected, you can START the program. Before STARTING though, you must determine if the program will be running synchronously or asynchronously, and if running asynchronously, you must also determine what the Reload Time will be. The Reload Time is the time between each successive execution of the C Program. This is programmable so that you can make the unit most efficient by telling the SCADA 3000 to execute the program only when necessary. For instance, if you require the program to run every half hour, it is not necessary to waste SCADA 3000's time and have the program run every 10 seconds.

The Reload Time must be at least *double* the Run Time and can only be programmed in whole numbers. The Run Time is provided when you RUN the program. So if your Run Time is 1.5 seconds, the Reload Time must be at least 3 seconds.

After the program is STARTED, SCADA 3000 will run the program depending on your programmed Reload Time (Async. mode) or everytime your ladder program runs (Sync. Mode).

Synchronous or Asynchronous

You can have your C program run sequentially with your ladder program (synchronous) or you can have it run independently on a fixed time interval (asynchronous). It depends on what your application requires. In synchronous mode, the inputs are scanned before the program starts and the outputs will not update until the C program has finished executing. In asynchronous mode, inputs and outputs are updated as changes occur during program execution. Generally, if you have a long C program or it takes a long time to run, you do **NOT** want it to run synchronously because this will slow your ladder program & input/output scanning to unacceptable levels.

Order of events in **synchronous** mode

1) Read motherboard inputs
2) Read expansion module inputs (if any)
3) Run ladder program (if enabled)
4) Run C program (if enabled)
5) Update main board outputs
6) Update expansion module outputs (if any)

Control Switch

On the front panel of the SCADA 3000 is a switch labeled "CNTRL". This switch is used to manually deactivate any control programs (Ladder, C program, PID control) onsite. If this switch is in the DISABLE position, no control programs will run. Additionally, you can have the control switch turn all outputs OFF. See the section on the Ladder & C-Program Start/Stop form ("Running and Stopping Ladder Execution") in Chapter 15.

VIEWING THE C VARIABLES

From the Main Programming form, choose *Programming>Variables*, or click on the *Ladder and C Variables* Button. This will bring up the following form.

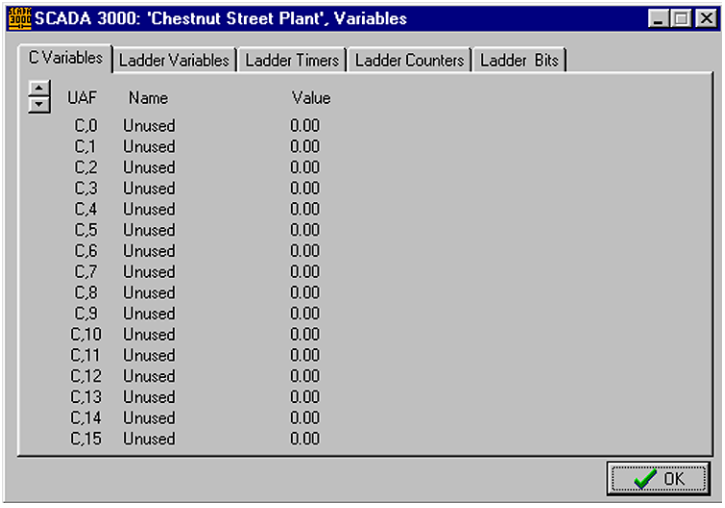


Figure 5: Variables Form

This form will show the real-time values for 16 C-variables at a time. Each variable UAF along with the variable name will be displayed. Use the Up/Down arrows to flip through the pages to access the variables you wish to view.

Forcing a Value

You can force C variable values for debugging purposes or to change the value of a constant. To change the value just double-click on the value you wish to change. The following form will appear:

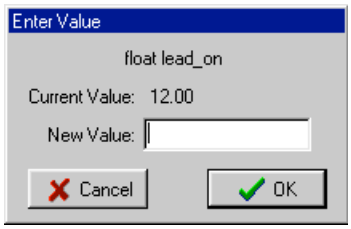


Figure 6: Value Form

Type in the new value in the box and click OK. The variable will immediately become the new value.

From the C Variables screen you can view all of your programmed variables in real time.

DIFFERENCES BETWEEN STANDARD C AND SCADA 3000 C

For those of you who are familiar with C programming, note that there are some differences between standard C and SCADA 3000's C compiler. The following items will be helpful to be aware of:

1) In SCADA 3000, every IF, ELSE, FOR, DO, and WHILE **must** have a set of brackets {} after it.

2) Condition clauses must be grouped together into pairs.

Standard C will allow:

```
IF ((condition 1) && (condition2) && (condition3))
```

SCADA 3000 requires:

```
IF (((condition1) && (condition2)) && (condition3))
```

3) There are no shortcut statements such as :

```
i++;
red+=5;   or
bits&=5;
```

However, statements such as the following are allowed if placed inside parentheses:

```
IF ((x=read_uaf(input,0,5))==0)
```

4) Bitwise operators are not implemented. There are no:

```
bitmask = bitmask &4;
```

5) Variables declared before the main statement retain their values after the program has been executed. This provides the programmer with nonvolatile memory between runs of the program. This is useful for accumulating, timing, and most SCADA 3000 applications.

6) There are no user defined functions or procedures.

7) Suggested programming philosophy:

Avoid the use of WHILE loops. This will lengthen the execution and response time of your C program and serves no advantage. Write your programs such that they run 'straight through' and exit. If your program is checking for a certain condition to occur, using a WHILE loop will cause your program to concentrate on that one condition unnecessarily. Since the program will execute at regular time intervals, you can use an IF statement and achieve the same or even better results.

For example, we need a program to set output 3 on whenever input 1 closes. Listed below is an example of how this would be done using a WHILE loop:

```
main ()
     {
     if (read_uaf(input,0,1)==0)
{
   write_uaf(output,0,3,on);
   while (read_uaf(input,0,1)==0);
              {
         }
          write_uaf(output,0,3,off);
          }
          }
```

In this program, when input 1 closes, output 3 will be turned on. The program will hold output 3 on as long as input 1 is closed. Unfortunately, this program will get stuck in the WHILE loop, waiting for input 1 to open. Critical actions may be missed because the program was forced to wait.

Listed below is the same program written using a 'straight through' approach. Only IF statements:

```
main ()
     {
     if (read_uaf(input,0,1)==0)
{
   write_uaf(output,0,3,on);
   }
        if  ((read_uaf(input,0,1)==1)
                {
         write_uaf(output,0,3,off);
   }
        }
```

In this case, when input 1 closes, output 3 will be turned on and then continue with the rest of the program. If input 1 opens, output 3 will be turned off. This program cannot get held up in a WHILE loop. Please note that these examples are greatly simplified for this discussion and are only intended to demonstrate the dangers that are possible when using WHILE loops.

C PROGRAMMING EXAMPLES

Example #1 Checking for alarms

The following lines of code will check to see if Alarm 9 exists.

```
If (read_uaf(alarm,9,0) != 0)
    {
    /* an alarm exists on alarm channel 9 */
    }
else
    {
    /* no alarm on channel 9 */
    }
```

Example #2 Checking for low-level alarms

The following lines of code will check to see if a low-level alarm exists on Alarm channel 9.

```
If (read_uaf(alarm,9,0) == 2)
    {
    /* a low-level alarm exists on alarm channel 9 */
    }
else
    {
    /* no low-level alarm on channel 9 */
    }
```

Example #3 Checking for high-level alarms

The following lines of code will check to see if a high-level alarm exists on Alarm channel 9.

```
If (read_uaf(alarm,9,0) == 3)
    {
    /* a high-level alarm exists on alarm channel 9 */
    }
else
    {
    /* no high-level alarm on channel 9 */
    }
```

Example #4 Forcing Alarms from the C-Program

To make an alarm occur as the result of a C-program action, you must define a C variable and then use the same C Variable as the Source Data Point on the Alarms form. On the alarms form you would select the Alarm Type to match the value of the C variable. For numeric values, set the Alarm Type to limit, for digital values set the Alarm Type to either bit=0 or bit =1. For example, suppose you have a C program that computes the average temperature of a liquid based on readings from several sensors. The average temperature is stored in a program variable called avg_temp. This C Variable would be used as the Source Data Point for an alarm and you would program high and low alarm limits accordingly.

```
float avg_temp;
float sensor_A;
float sensor_B;
float sensor_C;
float sensor_D;
main ()
{
   sensor_A=read_uaf(Input,0,10);
   sensor_B=read_uaf(Input,0,11);
   sensor_C=read_uaf(Input,0,12);
   sensor_D=read_uaf(Input,0,13);
   avg_temp = ((sensor_A + sensor_B + sensor_C + sensor_D)/4);
}
```

Example #5 Acknowledging Alarms from the C Program

The following lines of code show how to acknowledge an alarm from the C Program. If a momentary normally open push-button was connected to input zero, depressing the button would acknowledge the alarm.

Example:

```
   main ()
      {
      if (read_uaf(input,0,0)==1)      /*If input 0 on main board
is closed  */
         {
         reset (0);         /* then acknowledge alarm 0  */
         }
      }
```

Example #6 Lead-Lag Pump Down Control

This program will perform simple pump down control with alternating lead and lag pumps. The program controls two relay outputs from the reading of one analog level indicator. The level input is assumed to be on channel zero of the motherboard, and the relay outputs are assumed to be on channels zero and one of the motherboard. As long as those I/O points are acceptable, the only set up is to change the variable set up for 'lead_on', 'lag_on', and 'all_off'. For this example the three set points are preset in the source code. With some minor modifications you can make these parameters programmable from the C Variables form.

```
int lag;
int lead;
int state;
float lead_on;
```

```
float lag_on;
float all_off;
int first;
/* 0 = empty, 1 = lead on, 2 = lag on */
main()
  {
  if (first == 0)
    {
    lead = 0;  /* preset channel 0 as lead */
    lag = 1;   /* preset channel 1 as lag */
    lead_on = 12;
    lag_on = 15;
    all_off = 6;
    state = 0;
    first = 1;
    }
  if (state == 0)
    {
    if (read_uaf(input,0,0) > lead_on)
      {
      write_uaf(output,0,lead,on);
      state = 1;
      }
    }
  if (state == 1)
    {
    if (read_uaf(input,0,0) > lag_on)
      {
      write_uaf(output,0,lag,on);
      state = 2;
      }
    if (read_uaf(input,0,0) < all_off)
      {
      write_uaf(output,0,lead,off);
      state = 0;
      if (lead == 0)
        {
        lead = 1;
        lag = 0;
        }
      else
        {
        lead = 0;
        lag = 1;
        }
      }
    }
  if (state == 2)
    {
    if (read_uaf(input,0,0) < all_off)
      {
      write_uaf(output,0,lead,off);
      write_uaf(output,0,lag,off);
      state = 0;
      if (lead == 0)
```

```
        {
        lead = 1;
        lag = 0;
        }
      else
        {
        lead = 0;
        lag = 1;
        }
      }
    }
}
```

Chapter 17: PID PROGRAMMING

The SCADA 3000 includes eight programmable PID (Proportional, Integral, Derivative) control algorithms which allow the unit to operate as a feedback controller. A feedback controller is typically used to maintain a set point for a process. In a feedback control application you would have one input (the feedback) and one output (the force or mechanism) that allows you to maintain the setpoint.

To program a PID control algorithm, click on the PID button from the SCADA 3000 toolbar or select PID from the Programming menu. The following screen will be displayed.

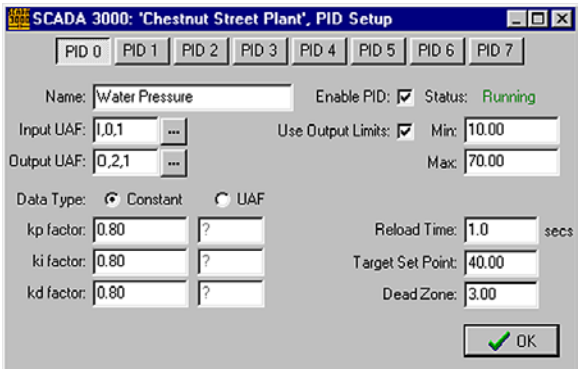


Figure 1: PID Form

Select the PID algorithm you wish to program by clicking on one of the buttons labeled *PID 0* through *PID* 7. Listed below are the programming parameters associated with each PID algorithm. Each parameter *must* have a value entered for that algorithm to calculate properly:

- **Name:** Type in a name to describe the process that the PID algorithm will be controlling. Example: Water Pressure.

- **Enable PID:** Clicking this box will start the PID algorithm. **Do not click it until you have finished programming all of the parameters associated with the PID form**. Once enabled, the algorithm will execute automatically based on the Reload Time. The Output UAF will change based on the results of the algorithm.

- **Input UAF:** The input UAF is the feedback for the process. This must be an analog signal. Select the input UAF by clicking the button next to the Input UAF text box. A choice of all possible UAF's will be displayed. Select the input which will provide the feedback for your process or enter the UAF manually by typing the UAF directly into the text box.

- **Output UAF:** The Output UAF identifies the output device which will affect the process. The output value will change as necessary to maintain the setpoint.

- **Reload Time:** The reload time determines how often the PID Algorithm is executed. For example, if the reload time is set to 0.25 seconds, every quarter second the PID algorithm will run.

- **Target Set Point:** This is the value that the PID algorithm is trying to maintain. The units of measure for the Target Set Point will be the same as for the Input UAF. For example, if the Input UAF is measuring pressure in PSI, then the Target Set Point must also be in PSI.

- **Dead Zone:** This is the amount of variation that can be tolerated around the Target Set Point. For example, if the you were trying to maintain a pressure of 75 PSI +/- 10, you would enter a value of 10 for the Dead Zone. This will permit the pressure to vary between 65 and 85 PSI without effecting any correction. Note that entering a value of 0 for the Dead Zone will cause the output to change frequently, which may be undesirable with certain types of equipment.

  NOTE: The PID Tuning factors below—Kp, Ki, Kd—must **all** be either constants or UAF locations, and never a mix of the two.

- **Kp factor:** This is the proportional tuning factor used in the PID algorithm. It may be either a constant or a UAF value.

- **Ki factor:** This is the integral tuning factor used in the PID algorithm. It may be either a constant or a UAF value.

- **Kd factor:** This is the derivative tuning factor used in the PID algorithm. It may be either a constant or a UAF value.

- **Use Output Limits:** Clicking this box will prevent the control output from exceeding the programmed Min and Max parameters.

- **Min:** The minimum output level that will be computed if output limiting is enabled.

- **Max:** The maximum output level that will be computed if output limiting is enabled.

Output Limiting can be used to restrict the SCADA 3000's ability to drive the process to its extremes. The SCADA 3000 PID algorithm also includes Integral term limiting to prevent the Integral from getting too large when the output reaches its limits. When Output Limiting is enabled and the output reaches the Min or Max value, the algorithm stops calculating the integral sum until the output comes back in range. Similarly, when Output Limiting is not enabled, the SCADA 3000 uses the output Table Low and Table High values to determine when to stop calculating the integral sum. This method of limiting the integral term is sometimes referred to as anti-reset windup.

Note that the Control Switch must be Enabled in order for the PID algorithms to run. In addition, the control switch can be programmed to turn all outputs OFF (including analog outputs) when the switch is placed in the Disable position. See the section "Running and Stopping Ladder Execution" in Chapter 15 for information on setting this option.

PID ALGORITHM

The most important and difficult part of programming a PID control process is setting the PID tuning factors. These factors will determine when to adjust the output, how much to adjust it by, and for how long. There are numerous articles and texts available that provide detailed analysis on how to set these parameters. If you are new to setting up a PID control process you may want to research this topic further. For our purposes we will cover the basic concepts of PID tuning.

The PID algorithm is made up of three terms that are added together to produce an output. The three terms are the proportional, integral and derivative terms.

```
Controller Output  =  Proportional  +  Integral  +  Derivative
                           Term          Term           Term
```

The proportional term is computed by taking the difference between the current input value and the set point (commonly referred to as the *error*) and multiplying it by the Proportional tuning factor *Kp*. The higher the value of Kp the more influence this parameter has on the controller output.

The integral term is computed by taking the integral of the *error* over a recent time interval and multiplying it by the Integral tuning factor *Ki*. The higher the value of Ki the more influence this parameter has on the controller output.

The derivative term is computed by taking the derivative of the error signal and multiplying it by the Derivative tuning factor *Kd*. The higher the value of Kd the more influence this parameter has on the controller output.

When the tuning factors are set correctly the system will maintain the set point without taking too long and without excessive overshoot or undershoot. If the current *error* is large, has been sustained for some time, or is changing rapidly, the algorithm will attempt to make a large correction by generating a large output. Conversely, if the feedback has matched the set point for some time, the algorithm will leave the output value alone.

TUNING

While this sums up the way you set the PID control algorithms, it doesn't really express the most difficult element of PID—and that is tuning. Your objective is to set the *Kp*, *Ki*, and *Kd* factors so that their weighted sum produces a controller output that steers the process variable in the desired direction but in such a way as to eliminate any error.

Setting all three tuning constants to large numbers might seem to solve this problem by ensuring an aggressive response to changes; however such a setup will at the very least cause a wild pendulum of cause and effect, and at the worst actually move the process variable further from the setpoint than before. Setting the tuning constants too conservatively, on the other hand, might not allow enough correction to fix one error before a new one appears. The properly tuned controller exists somewhere between these two extremes. While it recognizes and corrects an error quickly and aggressively, it doesn't overcompensate and throw the system into an unending loop of too much and too little.

The thing to understand is that there is no single answer except to say that the best tuning of a PID controller will depend on how the process responds to that controller's corrective efforts. Processes that react instantaneously and predictably need no control feedback at all. If you turn on a wall switch, your ceiling light comes on. A set amount of electricity arrives to power the illumination. No feedback is required. However, the amount of electricity available for you to draw upon must remain relatively constant throughout the day or your lights and appliances will suffer from potentially problematic surges and blackouts as power rises and falls. In a perfectly tuned system, compensation would be instantaneous. As that isn't possible—the shifts aren't precisely predictable—you see occasional (and hopefully very brief) brownouts during the "lag" between the network's increased need for power and the controller's response to that need.

In a slow process with a long lag time before the process responds to the controller's efforts, the proportional term will come into play to keep the output high or low until the error is eliminated. This will cause a cumulative error over time and the integral term comes into play; in fact, in a slowly changing process, it will come to dominate. The controller generates output based on the history of accumulating errors, and the variable overshoots the setpoint, causing an error in

the opposite direction. If that integral tuning constant is not excessive, then this rebound error will be minimal, and the variation will decrease as this error is added to the original. The derivative term will add its share to the controller output then, based on the derivative of this oscillating error signal—the back and forth between uncorrected and overcorrected states. The proportional term will rise and fall as the error diminishes toward a steadier state.

If the lag time is brief, however, and the process reacts quickly to the controller's efforts, then the integral term plays less of a role in the controller's output, as the errors will be too short-lived for it to work. Instead, the derivative term will be larger due to the rapid changes.

Each term will share some importance in any controller's output; how much is determined by the behavior of the process being controlled. Determining the best mix for your application is the essence of tuning. For a slow process, the *Kd* constant might need to be large in order to accelerate the controller's reaction to a setpoint change. But that same *Kd* setting for a fast-acting process might cause wild fluctuation and potential damage.

Kp, *Ki*, and *Kd* tuning is not an exact science. Experienced control engineers develop an intuitive sense of how to mix these factors. They've most likely achieved this through repeated trial and error. You can do the same. Tweak your parameters and then watch how the controller handles the next error. If it eliminates the error expeditiously, then leave your settings alone. If it's either too conservative or too aggressive, then adjust one or more of the tuning constants to compensate, and watch how it handles the next error, and so forth. This is not easy, as changing one tuning constant will likely affect one or both of the others as the rate and degree of error and compensation change.

Chapter 18: GAS FLOW MEASUREMENT

The SCADA 3000 has the capability of measuring natural gas flow. Calculations for up to four meter runs can be performed simultaneously. The SCADA 3000 supports two methods of gas flow measurement: orifice plate and positive displacement (turbine meter). Note that gas flow measurement using positive displacement requires a Pulse Count Module (FGD-3020).

The computations used when measuring gas flow with an orifice plate are in accordance with A.G.A. Report No. 3. The computations used when measuring with a turbine meter follow A.G.A. Report No. 7. Both calculations incorporate A.G.A. Report No. 8, Supercompressibility compensation. The A.G.A. standards and respective revisions to which the SCADA 3000 is designed are listed below:

A.G.A. Report No. 3, Third Edition, August 1992.
A.G.A. Report No. 7, First Revision, November 1984.
A.G.A. Report No. 8, Second Edition, 1992.

CAUTION: The SCADA 3000 is NOT approved or classified as an intrinsically safe device. The SCADA 3000 should never be installed in hazardous locations. The use of intrinsically safe barriers may be required when connecting this unit to any equipment in a hazardous location.

ORIFICE PLATE

When measuring gas flow with an orifice plate there are three signals which must be input to the SCADA 3000:

STATIC PRESSURE (upstream): Usually from a 4-20mA transducer (PSIA).
GAS FLOWING TEMPERATURE: Usually from a 10K thermistor (degrees F).
DIFFERENTIAL PRESSURE: Usually a 4-20mA signal from the orifice meter (inches of water).

These signals can be connected to any input on the SCADA 3000. In addition, they can be any type of analog signal that is compatible with the inputs of the SCADA 3000.

POSITIVE DISPLACEMENT

When measuring gas flow using positive displacement there are three signals which must be input to the SCADA 3000:

STATIC PRESSURE (upstream): Usually from a 4-20mA transducer (PSIA).
GAS FLOWING TEMPERATURE: Usually from a 10K thermistor (degrees F).
DISPLACED VOLUME: This signal must be the pulse output from a positive displacement meter or turbine meter. It must provide a dry contact signal to a SCADA 3000 Pulse Count Module.

The *static pressure* and *gas temperature* signals can be connected to any input on the SCADA 3000. In addition, they can be any type of analog signal that is compatible with the inputs of the SCADA 3000. The *displaced volume pulse output* must be connected to a SCADA 3000 Pulse Count Module (FGD-3020).

To program the SCADA 3000 for gas flow, click on the gas flow button from the toolbar or select Programming>Gas Flow from the main menu. The following form will appear:

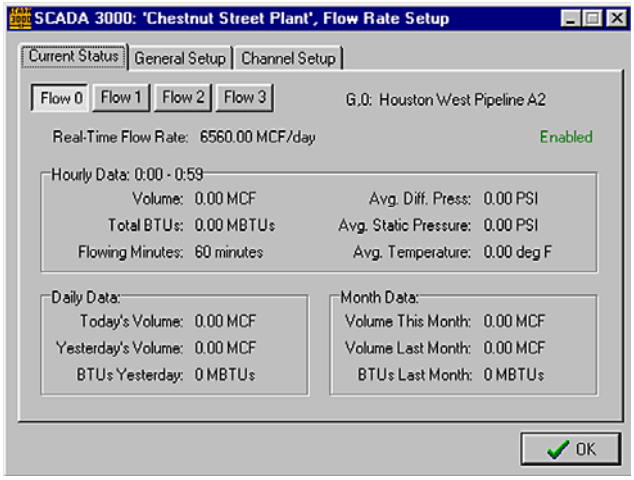


Figure 1: Gas Flow Form

This form has three tabbed screens, labeled *Current Status*, *General Setup*, and *Channel Setup*. The *Current Status* and *Channel Setup* tabs provide four buttons to select information from each of the four meter runs.

To begin your programming, start with *General Setup* by clicking on the *General Setup* tab. The following screen will appear:

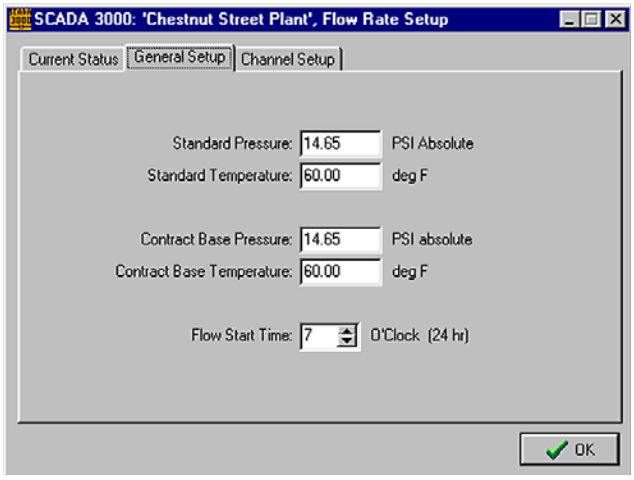

SCADA 3000: 'Chestnut Street Plant', Flow Rate Setup
Current Status
General Setup
Channel Setup
Standard Pressure: 14.65 PSI Absolute
Standard Temperature: 60.00 deg F
Contract Base Pressure: 14.65 PSI absolute
Contract Base Temperature: 60.00 deg F
Flow Start Time: 7 O'Clock (24 hr)
OK

Figure 2: General Setup tab

Enter the information for your gas flow equipment in the spaces provided. Some brief definitions and typical values are provided below:

Standard Pressure: 14.73 PSIA (typical)
Standard Temperature: 60 deg F (typical)

Contract Base Pressure (PSIA): Absolute static pressure of a gas at base conditions.

Contract Base Temperature (degrees F): Absolute temperature of a gas at base conditions.

Flow Start Time: Enter the Time of Day at which you want to start a new day's flow calculations. For example, if you enter 0, the daily calculations will start at midnight. Flow Start Time is set to a 24 hour clock. Thus, if you enter 23, flow calculations will begin at 11PM each day.

Once you have configured the *General Setup*, select the *Channel Setup* tab. The following form will appear:

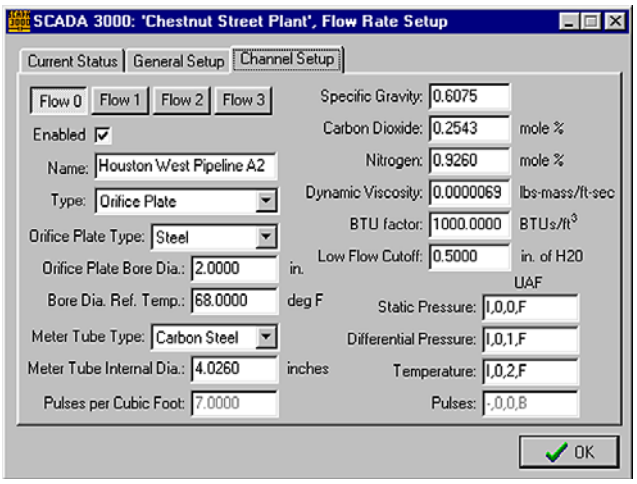


Figure 3: Channel Setup tab

This form has four buttons which are used to set up the four individual meter runs. If you are only measuring one meter run you do not have to set anything on the other Flow forms (e.g. Flow 1, 2, & 3). Begin by setting the parameters for your meter as listed on the form. The instructions below will guide you through each item.

Enabled: Checking this box will activate gas flow calculations for this meter run.

Note: Do not enable a meter run until all parameters are programmed and the required input signals are present.

Name: Enter a name to describe and distinguish each meter run. Up to 24 characters may be used.

Type (Orifice Plate/Positive Displacement): Select the type of meter you will be using.

Orifice Plate Type (Steel/Monel/Carbon Steel): Select the type of material used in the orifice plate (only used with Orifice type meters).

Orifice Plate Bore Diameter: Enter the bore diameter of the orifice plate. This figure should be available from the manufacturer's data (only used for orifice meter runs).

Bore Diameter Reference Temperature: Enter the reference temperature at which the bore diameter was measured. This figure should be available from the orifice plate manufacturer's data.

Meter Tube Type (Steel/Monel/Carbon Steel): Select the type of material used in the meter tube (only used with Orifice type meters).

Meter Tube internal diameter: Enter the meter tube internal diameter. This figure should be available from the manufacturer's data (only used for orifice meter runs).

Pulses per Cubic Foot: Enter the pulses per cubic foot for the positive displacement meter as provided by the meter manufacturer.

Specific Gravity: Enter the specific gravity (real gas relative density).

Carbon Dioxide Content: Enter the mole % of carbon dioxide (typical range: 0.04 to 28.94).

Nitrogen Content: Enter the mole % of nitrogen (typical range: 0.3 to 53.6).

Dynamic Viscosity: Enter the dynamic viscosity in pounds-mass per foot-second.

BTU Factor: Enter the factor to convert cubic feet of gas to BTUs.

Low Flow Cutoff: Enter the minimum differential pressure that determines when no gas is flowing. Units are in inches of H_2O.

Static Pressure UAF: Enter the UAF of the input that is measuring static pressure for this meter run. The transducer must provide the signal in units of PSI Absolute.

Differential Pressure UAF: Enter the UAF of the input that is measuring differential pressure for this meter run. The transducer must provide the signal in units of inches of H_2O (only used for orifice meter runs).

Temperature UAF: Enter the UAF of the input that is measuring gas flowing temperature for this meter run. The units for this input must be set to degrees Fahrenheit.

Pulses UAF: Enter the UAF of the input that is measuring pulses from a positive displacement meter for this meter run (only used for positive displacement meters; UAF must be an input from a Pulse Count Module).

Definitions

MCF: Thousand cubic feet

MCF/d: Thousand cubic feet per day

MBTUs: Thousand BTUs (British Thermal Units)

PSIA: Pounds per square inch absolute

CALCULATED INFORMATION

The SCADA 3000 will calculate and display instantaneous and accumulated data for each meter run. In addition, this data can be stored in the unit's datalogger by entering the UAF for the value you want logged. See the Datalogging section of Chapter 11 for more information. To view the calculated values click on the *Current Status* tab. The following form will appear:

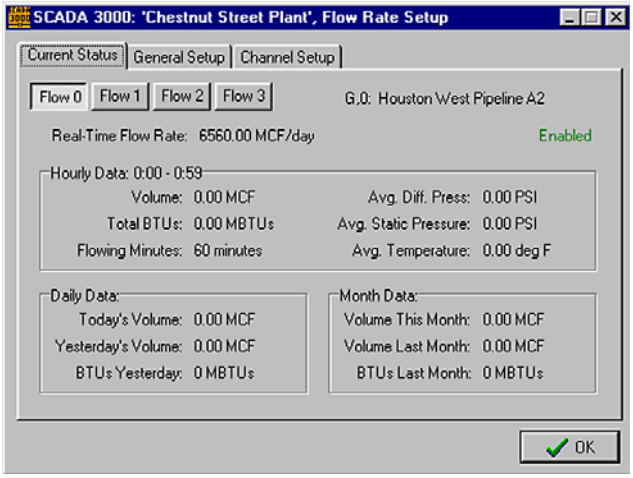


Figure 4: Current Status tab

Listed below are descriptions of the data, units of measure, and frequency of calculation for each parameter shown on the Current Status tab.

Real-time Flow Rate (MCF/d): Volumetric flow rate computed once per minute.

Hourly Data (updated on the hour)

Volume (MCF): Total volume of gas flow over the last full hour.

Total BTUs (MBTUs): Total BTUs over the last full hour.

Flowing Minutes: Total number of minutes of gas flow over the last full hour. Gas is determined to be flowing if the differential pressure is greater than or equal to the Low-Flow Cutoff (see *Channel Setup*).

Average Differential Pressure (in. of H_2O): Only averaged during flowing minutes (last full hour).

Average Static Pressure (PSIA): Only averaged during flowing minutes (last full hour).

Average Temperature (deg F): Average gas temperature over the last full hour.

Daily Data

Today's Volume (MCF): Total volume of gas flow so far today (updated on the hour).

Yesterday's Volume (MCF): Total volume of gas flow for the previous day.

BTUs Yesterday (MBTUs): Total number of BTUs for the previous day.

Note: The day starts based on the Flow Start Time entered on the General Setup tab.

Monthly

Volume This Month (MCF): Total volume of gas flow so far this month (updated daily).

Volume Last Month (MCF): Total volume of gas flow for the previous month.

BTUs Last Month (MBTUs): Total BTUs for the previous month.

Chapter 19: SYSTEM DIAGNOSTICS

MAIN BOARD DIAGNOSTICS

The SCADA 3000 includes circuitry to monitor numerous internal system parameters to make sure that the unit is functioning properly. Some circuits are tested on power-up while others are monitored on a continuous basis. Whenever a monitored function goes beyond its specified limits the *Fault* light is lit and a description of the problem is logged in the SCADA 3000's *Event Logger*. Faults can also be viewed on the LCD display. ON the Keypad, select MENU>DATA POINTS>DIAGNOSTICS>SYSTEM FAULTS. Realtime diagnostic information can be viewed by selecting *Diagnostics* from the main menu and then clicking on *Processors* or *Motherboard*. These will display system parameters monitored within the SCADA 3000 Main Unit. The two screens shown below:

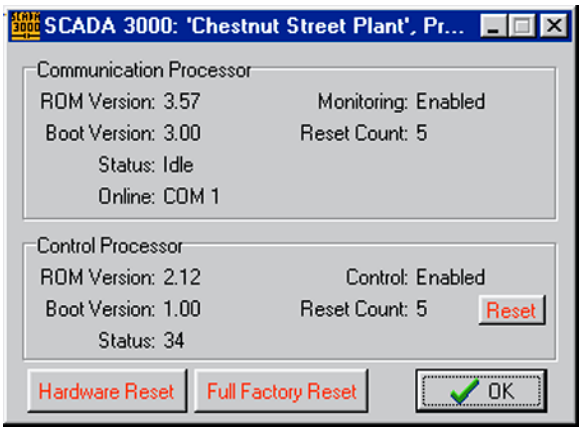


Figure 1: Processors Diagnostics

In the event of abnormal behavior, several RESET options can be performed on the unit's microprocessors. Within the Control processor group box is a *Reset* button which will issue a hardware reset to the control processor when clicked. This reset will not destroy programming. At the bottom of the Processor's Diagnostic screen are two additional reset options. The button labeled *Hardware Reset* will reset both the Communications and Control processor. This type of reset will not destroy programming; however, it will drop your connection if done while online using the modem. The *Full Factory Reset* will destroy all programming and restore the unit to factory default settings (new out-of-box). Do not click this button unless you are absolutely sure you have all programming backed up, including: SCADA 3000 unit programming, Ladder program, C-Program, DataLogger downloaded, and the Event log downloaded.

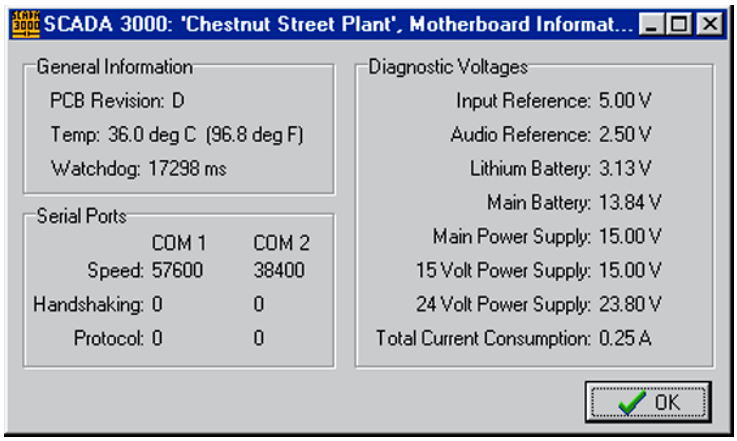


Figure 2: Motherboard Diagnostics

Between them, these screens provide status information for both microprocessors, the software version each is running, the circuit board temperature, numerous internal voltages, and system current consumption. Values shown are typical.

MODULE COMMUNICATIONS DIAGNOSTICS

When external I/O modules are connected, diagnostic information is provided to evaluate the communications performance between the Main Unit and each module. Click *Diagnostics* and select *Module Information*. The following screen will appear:

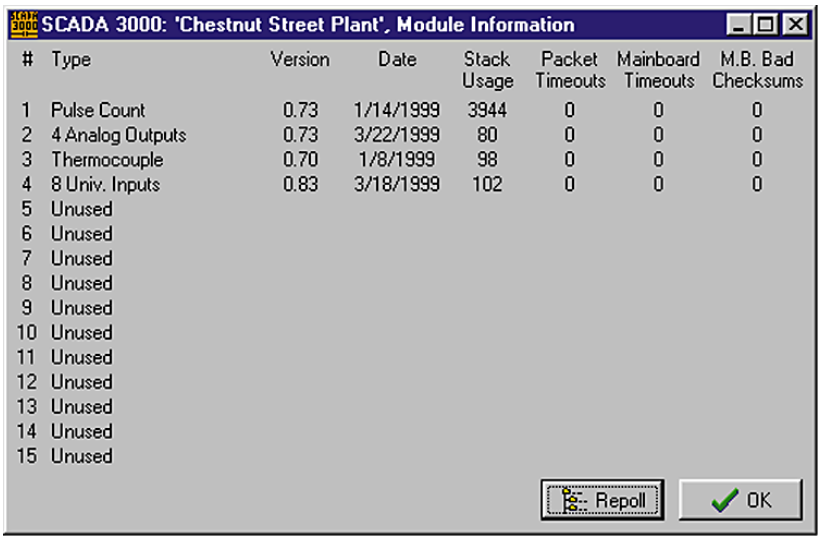

SCADA 3000: 'Chestnut Street Plant', Module Information
Type Version Date Stack Usage Packet Timeouts Mainboard Timeouts M.B. Bad Checksums
1 Pulse Count 0.73 1/14/1999 3944 0 0 0
2 4 Analog Outputs 0.73 3/22/1999 80 0 0 0
3 Thermocouple 0.70 1/8/1999 98 0 0 0
4 8 Univ. Inputs 0.83 3/18/1999 102 0 0 0
5 Unused
6 Unused
7 Unused
8 Unused
9 Unused
10 Unused
11 Unused
12 Unused
13 Unused
14 Unused
15 Unused
Repoll
OK

Figure 3: Module Diagnostics

The information on this form can help to identify communications problems due to wiring or interference. It also provides the software version and internal processor statistics as well. A well connected system would have zero errors (Timeouts & Bad Checksums) on the form above.

Chapter 20: INTERNET WEB STATUS

The SCADA 3000 Software can generate a web page and post it on the internet at a programmable time interval. The web page(s) will include alarm and selected I/O information for one or more SCADA 3000 units. The web page can be created in both standard HTML and WML formats; the latter allows viewing on an internet-capable cell phone. The web page can be updated on a programmable basis by setting up a polling schedule. In this scenario your host computer would contact one or more SCADA 3000's, retrieve the selected I/O status and alarm information, then create and upload the web pages to a specified server and directory on the world wide web hosted by your own internet service provider. Most internet service providers allocate space on their server for their subscribers to place their own web pages as part of the internet service contract. Contact your internet service provider for information. The Web Status software also allows you to save the pages to a local or network drive. This makes it possible to share the web page internally via a company intranet.

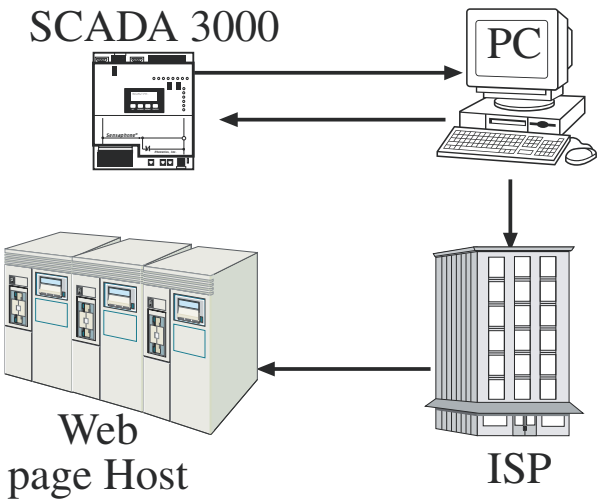


Figure 1: SCADA Web page path

REQUIREMENTS

This feature requires that you have a computer which is already set up to access the internet. Dial-up connections must use Microsoft Dialup Networking to get online. Providers which use proprietary software, such as America Online, cannot be used. Your internet service provider must support File Transfer Protocol (FTP).

INTERNET SETTINGS

Select your method of accessing the internet by selecting *File*, then *Options* from the Sensaphone Control Center. The following screen will appear:

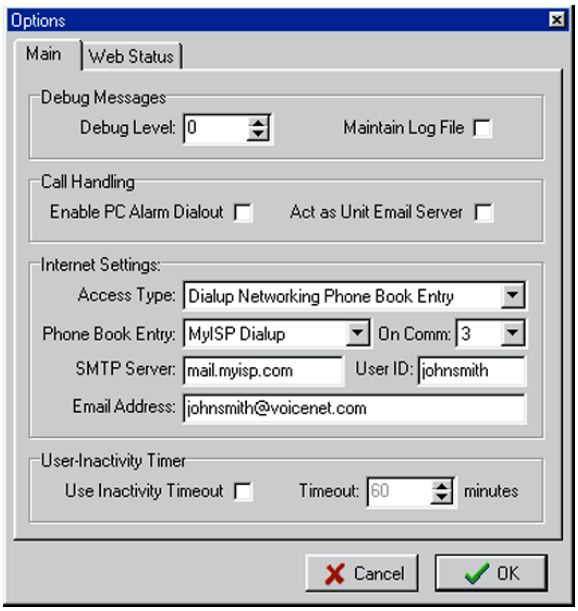


Figure 2: Control Center Options Main Tab

If your internet access is available all of the time, continuously, then choose Always On for your access type. If your internet access requires you to dial up using your modem, then choose Dialup Networking Phone Book Entry. Select the Phone Book Entry in the drop-down selection box that connects to your internet service provider. Next, select the *Comm* port that your modem uses to connect to the internet. (If you are not sure which Comm port your modem uses, then go to *My Computer*, double-click on *Dialup Networking*, right-click on the dial-up networking entry for your ISP and choose *Properties*. Next, click on the *Configure* button and locate the COM port information.)

> NOTE: You must have your internet account *Username* and *Password* saved in order for the SCADA 3000 software to automatically log in.

If your service provider does not use Microsoft Dialup Networking then no names will appear in the Phone Book Entry list and you will have to use a different internet service provider. Companies such as America Online, which use proprietary software, will not work with the Web Status feature.

GLOBAL WEB STATUS SETTINGS

Your web page will be uploaded to an internet server using FTP (File Transfer Protocol). Listed below are several parameters which are required for FTP to access and load your web page to the designated web server. Contact your internet service provider to determine the proper settings.

Host Name (location of your web site)

Remote Directory (folder within the web site where the pages will upload)

Username (your account information necessary to access the web server)

Password (your account information necessary to access the web server)

Click on the *Web Status* tab to bring up the following screen:

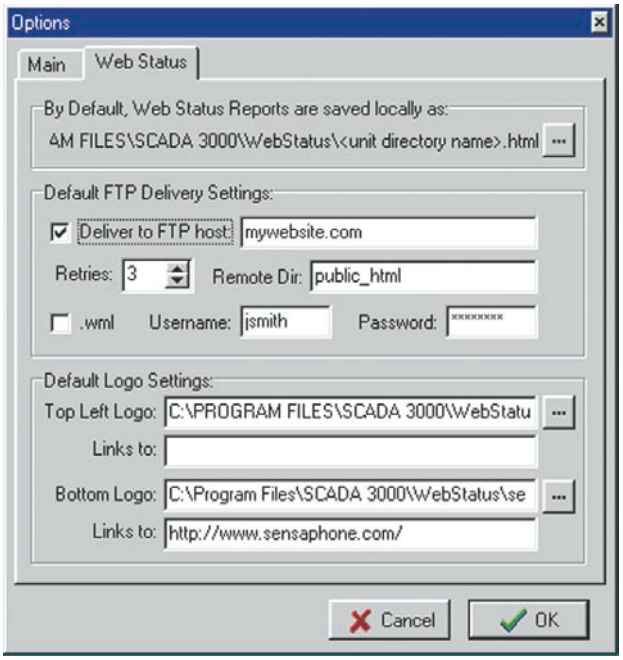


Figure 3: Global Web StatusTab

Check the "Deliver to FTP host" box to enable the delivery settings. The host address should be the ISP address of your site.

Retries is the number of times the software will attempt to upload new page data.

Remote Dir: is the folder or directory within your site where the web page is to be placed.

.wml: Check the *.wml* box if you would like web pages created in wml format in addition to the HTML format. The wml format is specifically for use with internet-capable cell phones.

Username and *Password* are those needed to access and make changes via FTP to your site.

> In addition to FTP programming, the screen also allows you to set default logos for your web page. The default settings shown above provide a Phonetics logo graphic at the top of the page and a Sensaphone banner graphic at the bottom. The bottom graphic will link to www.sensaphone.com. A Logo or picture file (JPEG, GIF, PNG) of your choice can be selected to appear at the top and bottom of your web page rather than the default graphics. You can also set up the logos to link to web pages of your choice.

The settings here will operate as the global default settings for all of your SCADA 3000 units. The chosen host and directory are where all web status reports will appear. The file name of the web page will be based on the SCADA 3000 unit name, such as "0_ChestnutStreetPlant.html"; however, this can be customized to your preference. Note that the default settings can be overridden on a per-unit basis, making it possible to upload web pages for individual units to different web servers. (See "Web Status Options".)

SELECTING DATA POINTS FOR WEB STATUS POLLING

Go online with the SCADA 3000 unit that you want to configure for web status polling. On the Main Programming Form, select *Status*, then *Web Status Report* to bring up the Web Status form. Click on the *UAFs* tab.

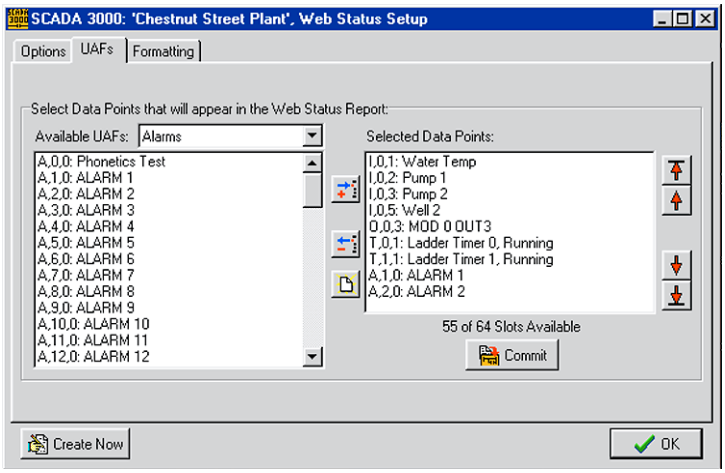


Figure 4: Web Status UAFsTab

All data points in the SCADA 3000 are referenced by the UAF format. On the form above there are two large list boxes, *Available UAFs* and *Selected Data Points*. *Available UAFs* comprises all the data points that can be chosen to appear in the status report. *Selected Data Points* on the right shows which ones have been selected. From the drop down box next to the heading *Available UAFs*, choose the type of value you wish to add—Inputs, Outputs, Timers, Counters, Bits, Ladder Variables, C Variables, Gas Flow, Alarms, and Diagnostics. When you pick a category, a list of all possible points in that category will appear. Click on the item you wish to add to the Web Report and then click on the first of the three buttons between the two list boxes, the one with an arrow pointing to the right. This will copy the highlighted item to the list of *Selected Data Points*. Continue this process by highlighting each chosen data point and copying it to the list of *Selected Data Points*. To remove a data point from the list, highlight the data point from the *Selected Data Points* list box and click on the middle button, the one with the arrow pointing towards the left. This will remove the highlighted item.

When you are finished editing the list, click on the *Commit* button. Next select the *Formatting* Tab.

FORMATTING THE WEB PAGE

The web page that is created can be programmed to refresh automatically, updating the data, at specific intervals. It will also contain a top and bottom Sensaphone logo which can be changed based on your preferences. In addition, you can have the logos be active links to other world wide web addresses. The *Formatting* tab is where you specify these elements of your page.

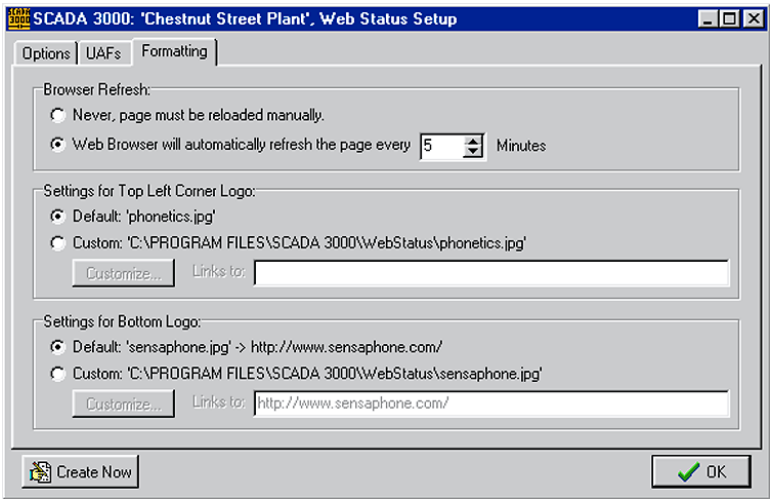


Figure 5: Web Status Formatting tab

Browser Refresh

If you select "Never, page must be reloaded manually", the data on the Web Status page will not change automatically, but only when you manually reload it.

If you select "Web Browser will automatically refresh the page every *X* minutes", the browser will update the web page at the interval in minutes that you specify. Note that this is independent of the actual polling interval over which new SCADA 3000 data is being collected. If the Scada 3000 hasn't been polled, then the web page isn't going to change and, as a result, the refresh will simply reload the old page. It is recommended that you set the web page refresh to a time interval that matches the polling interval.

Settings for Top Left Corner Logo

"Default: 'phonetics.jpg'" will place the Phonetics logo in the top lefthand corner of the page.

Select "Custom" and click on the *Customize* button if you want to select a logo of your choice. The file must be in one of the following formats: JPEG, GIF, or PNG. Type an http address in the "Links to:" box if you want the logo to perform as a live link on the status page.

Settings for Bottom Logo

Similarly, the default for the bottom logo is the "sensaphone.jpg" banner. Select "Custom" to replace that banner with a graphic of your choice, and add an http address in the "Links to:" box if you want the graphic to act as a live link.

Once you've finished formatting the page, click on the *Options* tab.

WEB STATUS OPTIONS

Check the "Enable Web Status Report" box to generate web status pages for this unit.

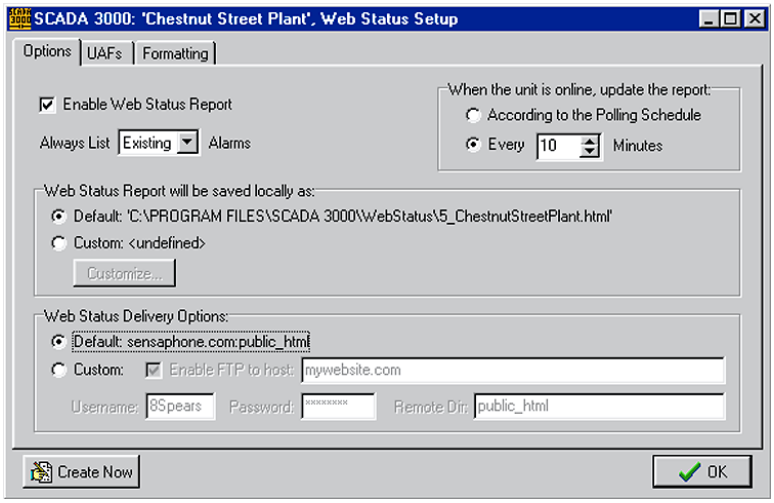


Figure 6: Web Status Options tab

Alarm Options

In addition to displaying the values of I/O points, your web page can also include detailed alarm information. Three options are available to customize how you would like alarm information to appear.

ALL will list the status of all the enabled alarms regardless of whether or not the alarm exists or has been acknowledged.

NO will list no alarms in the web status report.

EXISTING will list only those alarms that have been activated at the time of the report. If there are no current alarm conditions, none will show.

Online Options

When the SCADA 3000 is online, you can have a web page generated and uploaded according to the polling schedule you've set up (see "Setting up a Polling Schedule for Web Status"), or you can select a time interval of your choosing by enabling the "Every *x* minutes" option. When this option is selected a new web page is created and uploaded based on the time interval.

Local Web Pages & Web Page Filenames

Whenever a new web page is created, the software will save a copy locally on your hard drive in addition to the one that is uploaded. You can optionally change where the local copy is saved so that it can reside on a network server or Intranet server. To change where the local copy is saved, click the *Custom* option and then click the *Customize* button to assign the web page a file location and name of your choice. Note that the filename you use will also be assigned to the file uploaded via FTP. WML web pages will be appended with the prefix "wap". For example, if the filename was specified as *Chestnut.html*, the wml filename would be *wapChestnut.wml.*

FTP Delivery Options

Web pages for the selected SCADA 3000 can be sent via FTP to the default location or optionally they can be sent to another location. Typically you will have all of your web pages hosted on the same internet server. When you program your FTP settings from the Sensaphone Control Center you set up a system default. However, you can optionally set up the software to customize the FTP location for each SCADA 3000 unit on an individual basis. Click the *Custom*

option and you can set up a separate FTP location so that only the web page for the selected SCADA unit will be uploaded to this location.

Instant Web Page

To create a web page immediately, click the *Create Now* button. This will make the SCADA 3000 software poll the unit for I/O and Alarm information, create the web page, and FTP the files to your ISP. Be sure to have all of your web page programming completed before clicking this button.

SETTING UP A POLLING SCHEDULE FOR WEB STATUS

From the Main Menu of the Sensaphone Control Center screen, select *Polling* and then *Schedule*. The current polling schedule will appear. Click the *Add Poll* button to create a new polling entry. The following screen will appear:

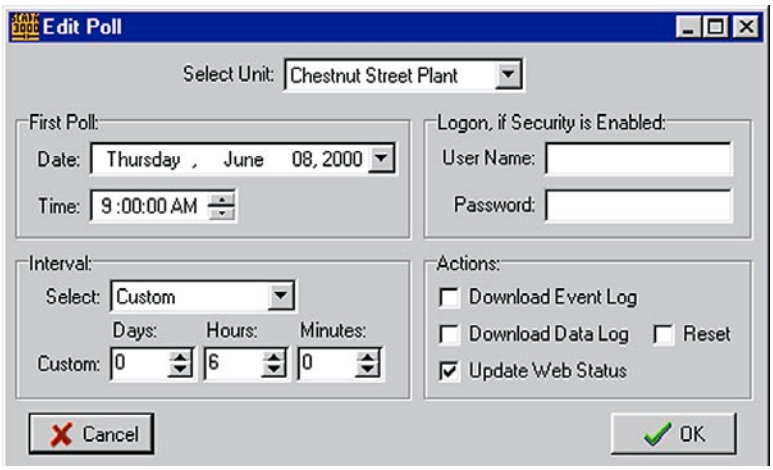


Figure 7: Edit Poll form

Setting up a polling schedule is detailed in Chapter 11. For the purposes of this section we will briefly cover the main points. Select a unit to poll using the selection box at the top of the screen. Set the time for the first poll and the interval for all subsequent polls. The interval can be set for monthly, weekly, daily or custom time intervals. The example above shows a custom interval set at 6 hours. Next, choose the action(s) to be performed during the poll. In this case you want to choose *Update Web Status.*

Note: You can select more than one action to take place during a poll.

As programmed above, the selected unit, Chestnut Street Plant, will be polled every six hours, and a new web page will be generated and uploaded to the internet. To create web pages for more than one unit, you must set up a separate poll for each. Finally, be sure to enable polling by clicking the *Polling Enabled* box on the Polling Setup screen.

FREQUENTLY ASKED QUESTIONS

Can I create web pages for more than one SCADA 3000?

Yes. Every time you poll a 3000 unit the software can create a new web page for that unit. Each unit will have its own web page address based on the unit name. If you repoll the same unit, the software simply overwrites the old web page with the new information.

Do I need a separate internet account for each SCADA 3000?

No. Generally you can put as many web pages on one account as you like. Internet providers allocate a certain amount of storage for web pages. Since the SCADA 3000 web pages are relatively small (60K) you can put many on one site.

Why can't I find my webpage?

Chances are you're just typing the wrong address or you're not entering the correct upper and lower case letters in your Username or web page Filename. Typically your webpage will be located at some subdirectory of the main ISP's address which will be based on your Internet Account Username. For example, if you had an internet account with Voicenet and your Username was JSmith, your webpage would be located at www.voicenet.com/~JSmith/#_unitname.html. Note that the unitname and Username are case sensitive. You can see what the web page filename will be by viewing the local copy settings on the web status programming screen.

How do I find out what Internet settings to use?

You can typically find this information in one of the following places:

(1) In the original internet account information provided by your ISP;

(2) On your ISP's website in the Tech Help, FAQ, or Support section; or

(3) Call or email your ISP for the required information.

Why can't I FTP a web page to my America Online account?

America Online requires proprietary software to upload files to their server.

Can I program my SCADA 3000 through the web page?

No. You can only view information on the web page.

Can I acknowledge alarms through the webpage?

No. You can only view alarm and input information on the web page.

How long does it take to poll for web status and deliver (FTP) the web page?

If your Internet access is via dialup, then the whole process takes about $2^1/_2$ minutes.

Chapter 21: EMAIL SERVER OPTION

The SCADA 3000 software can be used as an e-mail server to receive and forward e-mail messages from one or more SCADA 3000s. This means that if you have a computer with internet access, you can use it to receive e-mail calls from your 3000 and forward them over the internet yourself, instead of using the Phonetics pay service for e-mail. You can choose to use the same modem for both incoming and outgoing calls or you can add a separate modem specifically to handle additional calling activity. The choice to use one or two modems depends on how much dialing activity you have on your computer (SCADA 3000 polling, incoming alarms, e-mail reception, internet dial-up, etc.).

REQUIREMENTS

The E-mail Server feature requires that you have a computer which is already setup to access the internet. It can be either an always-on connection (via DSL, cable modem, T1) or a dial-up connection. Dial-up connections must use Microsoft Dial-up Networking to get online. Providers that use proprietary software, such as America OnLine, cannot be used. You must also have a telephone line and modem to receive the e-mail calls from your SCADA 3000(s).

HOW DOES IT WORK?

Each SCADA 3000 has a telephone number it calls when it delivers e-mail. Typically this is the telephone number of the modem connected to Phonetic's e-mail server; however, you can change this number. By changing the e-mail telephone number to your own modem, your computer becomes the e-mail server. When your SCADA 3000 needs to send a report or alarm message to an e-mail address, the unit will dial your computer and send the message. Your computer will then log on to the internet and forward the e-mail message to the designated party.

PROGRAMMING

To set up your computer as an e-mail server, click *File*, then *Options* from the SCADA 3000 Control Center Menu. The following screen will appear:

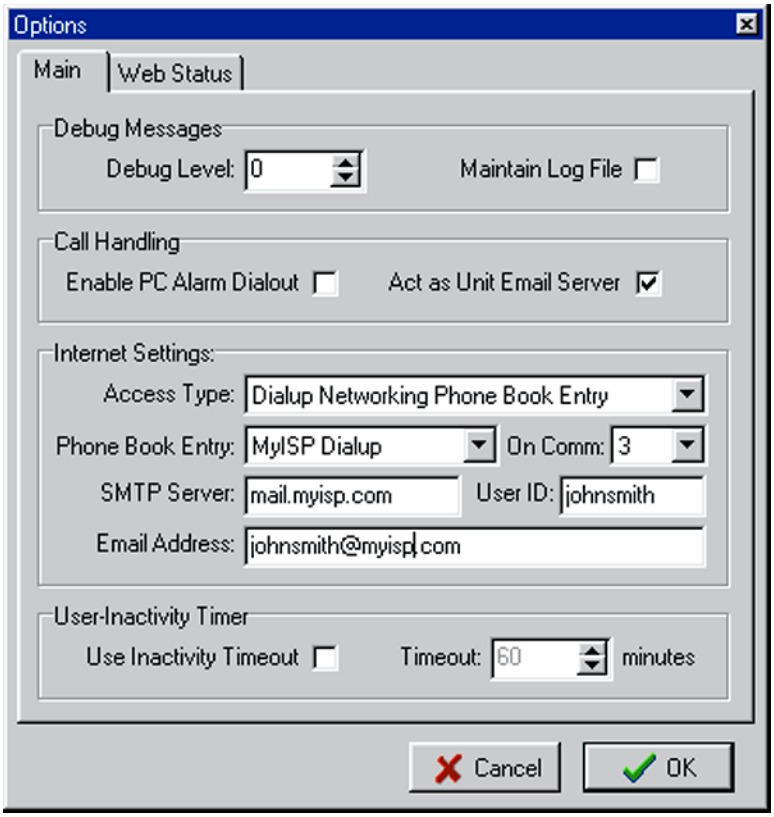


Figure 1: Control Center Options Main Tab

In the Call Handling section, check the box labeled *Act as Unit Email Server*. Under Internet Settings, select the type of internet access for your computer (either *Dialup Networking* or *Always-On*). If you chose Dialup Networking then select a Phone Book Entry by clicking the Down arrow. Next, select the Comm port that your modem uses to connect to the internet. (If you're not sure which Comm port your modem uses then go to *My Computer*, double-click on *Dialup Networking*, right-click on the dial-up networking entry for your ISP and choose *Properties*. Next, click on the *Configure* button and locate the COM port information.) Enter the name of the SMTP (Simple Mail Transfer Protocol) server used by your ISP (Internet Service Provider). This should be included with the account information from your ISP. You may also obtain it by calling them or checking their website. Next, Enter your e-mail account User ID as provided by your ISP and your e-mail address.

Next, select *Comm*, then *Setup* from the Sensaphone Control Center main menu. Select the Comm port tab corresponding to the modem that will be receiving e-mail calls. Click the box labeled *Answering Incoming Calls*, as shown below:

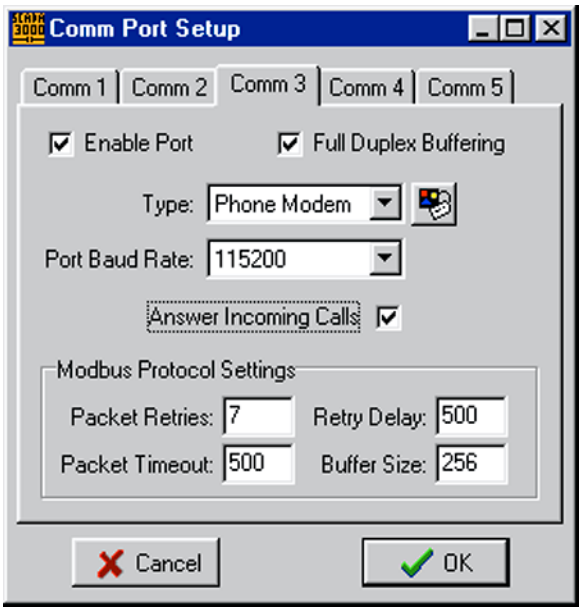


Figure 2: Comm Port Setup form

The last step is to change the e-mail telephone number in the SCADA 3000. Go on-line with each of your SCADA 3000 units and click on the *System* button from the toolbar or select *Programming*, then *System* from the Main menu. Next, Click on the *Advanced* button located within the Dialout Settings group. The following screen will appear:

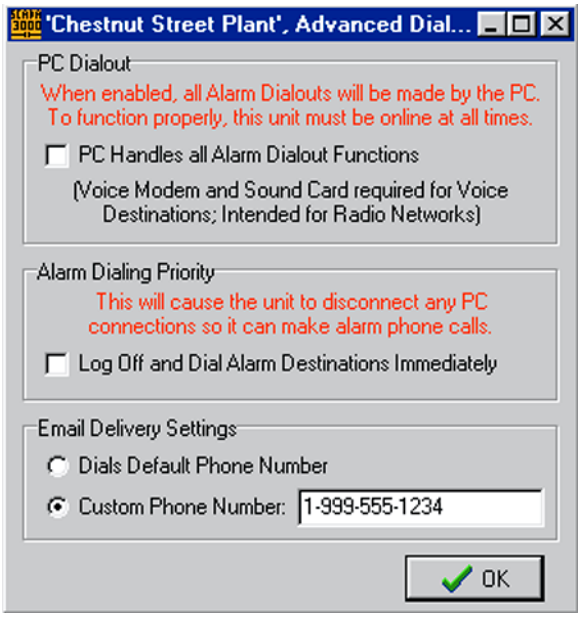

'Chestnut Street Plant', Advanced Dial...
PC Dialout
When enabled, all Alarm Dialouts will be made by the PC.
To function properly, this unit must be online at all times.
PC Handles all Alarm Dialout Functions
(Voice Modem and Sound Card required for Voice Destinations; Intended for Radio Networks)
Alarm Dialing Priority
This will cause the unit to disconnect any PC connections so it can make alarm phone calls.
Log Off and Dial Alarm Destinations Immediately
Email Delivery Settings
Dials Default Phone Number
Custom Phone Number:
1-999-555-1234
OK

Figure 3: Advanced Dialout form

Located at the bottom of the screen is the selection box to dial a custom phone number for sending e-mail. Click in the box and enter the telephone number of your computer modem which will be receiving the e-mail calls from the SCADA 3000. Click *OK*.

Chapter 22: PC ALARM DIALOUT

PC Alarm Dialout is a feature which allows your computer to make alarm calls (voice, fax, pager, email ...) when your SCADA 3000 units do not have access to a telephone line. The feature is intended to be used when you have several SCADA 3000 units connected via a radio network or serial port. When an alarm occurs on one of the units, the PC will make the alarm calls and receive the acknowledgement code.

Computer Requirements

Diamond Supra Express V.90 Speakerphone modem

Soundblaster Compatible Soundcard & Speakers

Microphone

*Cable with 1/8" male to 1/8" male connectors

*The cable must connect to the Line Out jack of the Soundcard and to the Mic input of the modem. The microphone plugs into the MIC jack of the soundcard. See figure below:

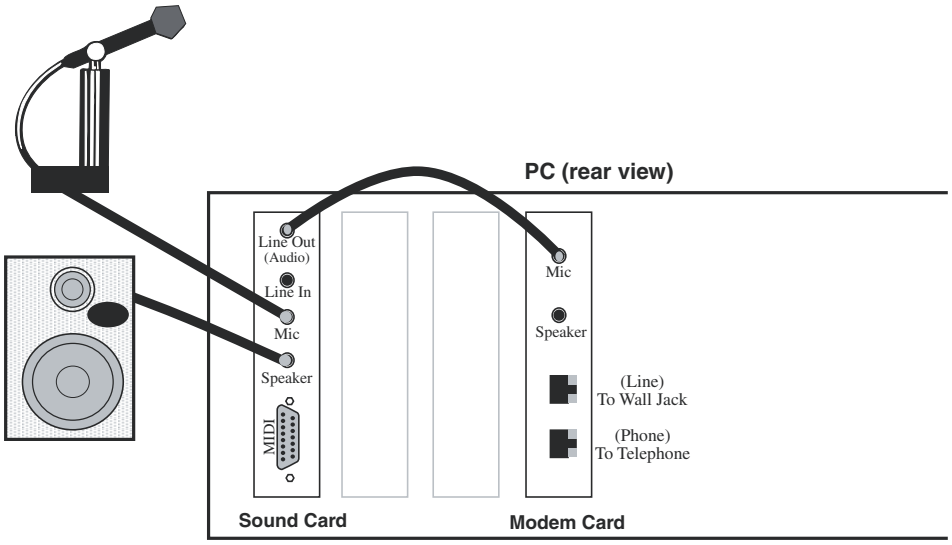


Figure 1: Typical connections between modem & soundcard

Note: If your sound card does not have a separate speaker out jack, you'll need to use a splitter cable to connect line out to both speakers and the Modem Card microphone jack.

SOFTWARE SETUP

The PC Alarm Dialout feature is enabled from the SCADA 3000 Control Center screen. Run the SCADA 3000 software and select *File*, then *Options*. In the *Call Handling* section, click in the box labeled *Enable PC Alarm Dialout*, then click *OK*. This will bring up the PC Alarm Dialout status window shown below:

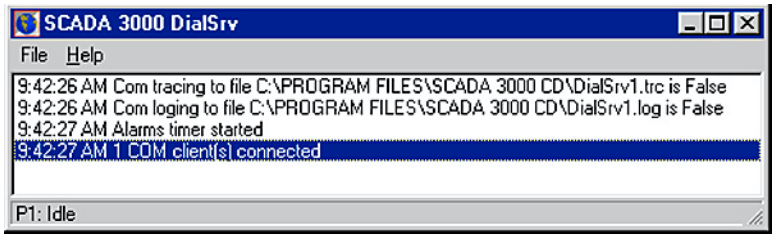

SCADA 3000 DialSrv
File Help
9:42:26 AM Com tracing to file C:\PROGRAM FILES\SCADA 3000 CD\DialSrv1.trc is False
9:42:26 AM Com loging to file C:\PROGRAM FILES\SCADA 3000 CD\DialSrv1.log is False
9:42:27 AM Alarms timer started
9:42:27 AM 1 COM client(s) connected
P1: Idle

Figure 2: PC Alarm Dialout status window

Click *File*, then *Setup* to bring up the software setup screen. (See figure below)

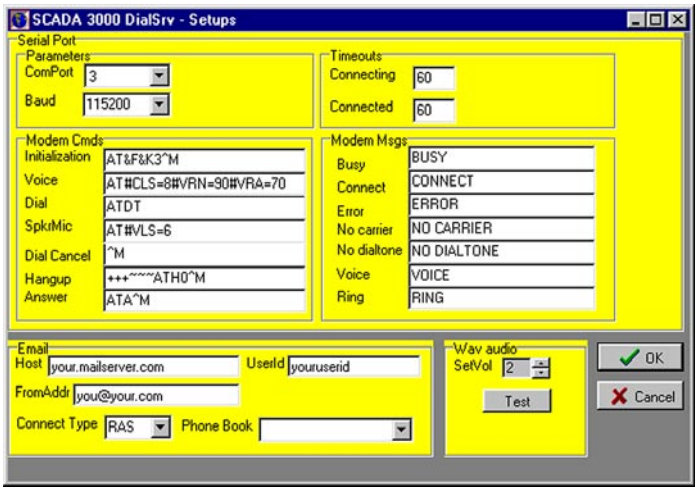


Figure 3: PC Dial Server Setup

In the Parameters section, select the Com Port where your modem is installed. Set the maximum baud rate supported by your modem (typically 115,200).

Connect Timeout: This is the amount of time (in seconds) allowed for a connection to be established. The timer starts when the number is dialed. (default = 60 seconds)

Disconnect Timeout: This is the maximum amount of time (in seconds) permitted for message delivery to take place after a connection has been established. (default = 60 seconds)

Modem Commands and Modem Messages

Do not change. These settings are configured to work with the Diamond Supramax Express Speakerphone modem. The ability to alter these settings is provided for cases where a different modem is being used. Contact Phonetics Technical support for more information.

Email Settings

The PC Alarm Dialout program can send alarm messages to any e-mail address if your computer has internet access. The program supports two methods of accessing the internet: RAS Dialup, or Local Always-On LAN connection (DSL, T1, Cable modem, etc.). Enter the following information:

Host: Enter the name of the SMTP server used by your Internet Service Provider (ISP).

UserID: Enter your e-mail account user ID as provided by your ISP.

FromAddr: Enter your e-mail address.

Connect Type: Select your method of accessing the internet by clicking on the dropdown arrow. Choose either RAS (dialup) or Local (always-on).

• RAS Dialup: If you connect to the internet using a dial-up modem, then select "Dialup (RAS)" from the *Type* menu. Select the service provider you want to use from the *PhoneEntry* pulldown menu. Most users will only have one service provider set up through Microsoft Dialup Networking (MDN), and so will only have one choice in the pulldown menu.

> **Note**: You must have your MDN internet account Username and Password saved in order for the PC Alarm Dialout program to automatically log in. Providers which use proprietary software, such as America Online, cannot be used.

• Local Always-On (LAN etc.): If your internet access is available all of the time, continuously, then choose "Local Always-on (LAN etc.)" for your access type.

Telephone Voice Volume Setting

The Wav Audio settings allow you to set the audio level when the computer makes a voice telephone call. Typically this setting should be around 1 or 2 depending on your soundcard and modem. Use the up/down arrows to increase or decrease the audio level.

SCADA 3000 PROGRAMMING

In order for the SCADA 3000 to use the PC Alarm Dialout feature you must configure each unit on the network as described in this section. For each SCADA 3000 unit you must do five things:

1. Enable PC Alarm Dialout
2. Program the ID number
3. Program voice messages
4. Program destinations
5. Program alarms

Step One is to let the SCADA 3000 know that the PC is going to handle all alarm dialing activity. To do this, log in to each SCADA 3000 and click the *System Programming* button from the toolbar or select *Programming*, then *System* from the main menu. On the System screen click the *Advanced* button, located within the *Dialout Settings* area. The following screen will appear.

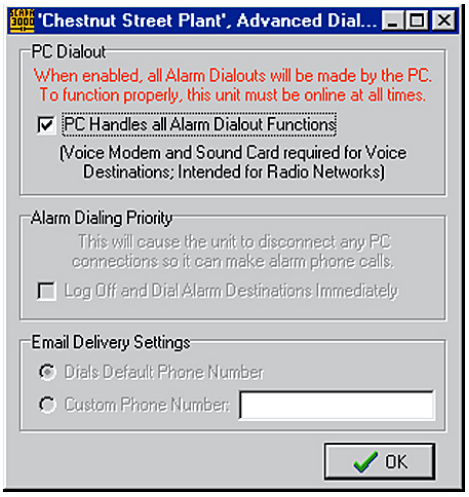

'Chestnut Street Plant', Advanced Dial...
PC Dialout
When enabled, all Alarm Dialouts will be made by the PC.
To function properly, this unit must be online at all times.
PC Handles all Alarm Dialout Functions
(Voice Modem and Sound Card required for Voice Destinations; Intended for Radio Networks)
Alarm Dialing Priority
This will cause the unit to disconnect any PC connections so it can make alarm phone calls.
Log Off and Dial Alarm Destinations Immediately
Email Delivery Settings
Dials Default Phone Number
Custom Phone Number:
OK

Figure 4: Advanced Dialout Screen

Check off the box labeled "PC handles all Alarm Dialout Functions". This will prevent the SCADA 3000 from trying to dial out when an alarm exists.

Note: PC Alarm Dialout does not support the delivery of Reports.

Step Two is to program the Unit ID Number which is located on the System screen. Typically this would be the telephone number of the SCADA 3000, but in this case enter the telephone number of the computer modem. By entering this number, personnel will receive the computer's telephone number on all alarm messages. This way, personnel responding to alarm calls will be able to dial back to the computer, which will allow them to acknowledge alarms.

Voice Messages

Step Three is to program voice messages. Click on the *Voice Programming* button from the toolbar or select *Programming*, then *Voice* from the main menu. The following screen will appear.

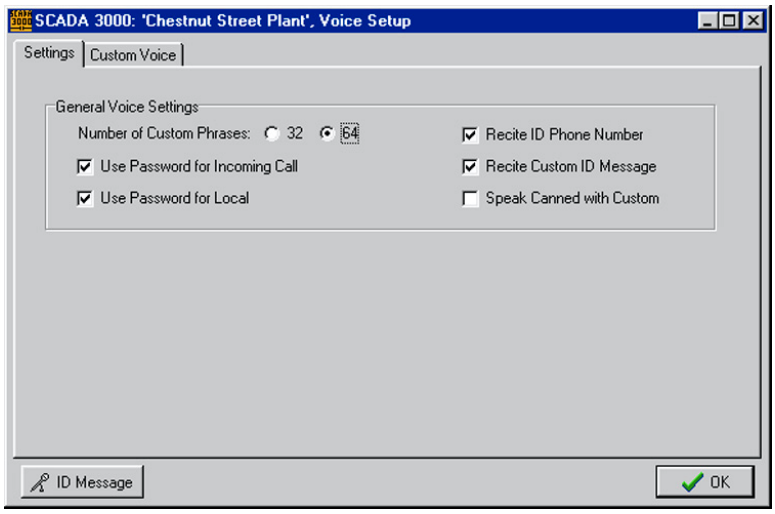


Figure 5: Voice Setup Screen

This screen can be used to select certain options when a voice alarm call takes place. The options are:

* Should the ID phone number be recited
* Should the custom ID message be recited
* Should canned messages be recited with custom messages

Select the options you want by placing a check mark in each box.

Next you must record voice messages. At the bottom of the screen above is a button labeled *ID Message*. Click this button to bring up the ID Message recording screen (see below).

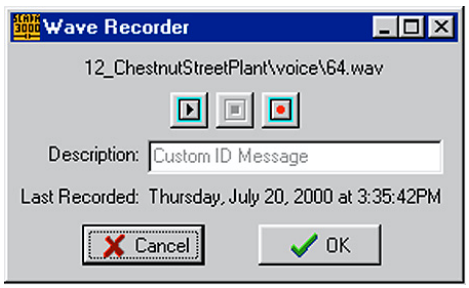


Figure 6: Wave Recorder screen

To record the ID Message, click the button with a red circle and speak into the microphone connected to your computer. Click the Center button to stop recording. To play the message

back, click the button with an Arrow. The ID Message is typically a message which identifies where the unit is located. For example: "This is the monitoring system for the Chestnut Street Water plant in Southern Brisco County."

Next, click on the Custom Voice tab at the top of the Voice Setup screen. The following screen will appear:

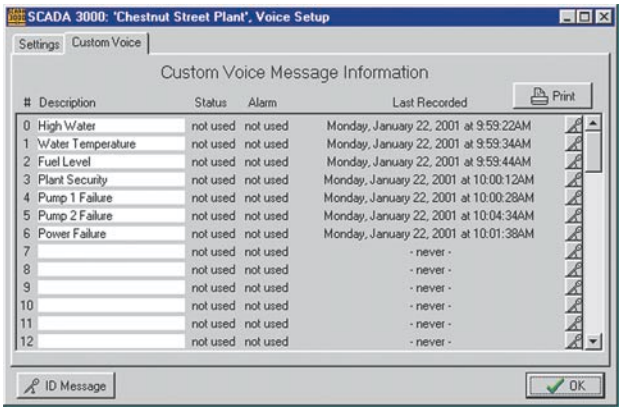

SCADA 3000: 'Chestnut Street Plant', Voice Setup
Settings
Custom Voice
Custom Voice Message Information
Print
Description Status Alarm Last Recorded
0 High Water not used not used Monday, January 22, 2001 at 9:59:22AM
1 Water Temperature not used not used Monday, January 22, 2001 at 9:59:34AM
2 Fuel Level not used not used Monday, January 22, 2001 at 9:59:44AM
3 Plant Security not used not used Monday, January 22, 2001 at 10:00:12AM
4 Pump 1 Failure not used not used Monday, January 22, 2001 at 10:00:28AM
5 Pump 2 Failure not used not used Monday, January 22, 2001 at 10:04:34AM
6 Power Failure not used not used Monday, January 22, 2001 at 10:01:38AM
7 not used not used - never -
8 not used not used - never -
9 not used not used - never -
10 not used not used - never -
11 not used not used - never -
12 not used not used - never -
ID Message
OK

Figure 7: Custom Voice screen

This screen shows the custom voice message descriptions, which alarm they are associated with, and when they were recorded.

> **Note**: Voice message association is programmed on the Alarms programming screen (Notification Tab).

The first step is to enter a voice message description for each message you will be recording. Next, use the buttons to the far right of the screen to record each voice message.

PROGRAMMING DESTINATIONS

Step Four is to program destinations. Destination programming is identical to the method used for normal operation for destination types except Numeric Pagers. The following differences apply:

Since the PC Alarm Dialout program can process alarms from multiple SCADA 3000 units at once, a unique Alarm ID number is added to every PC-generated alarm message. The Alarm ID number is automatically added to all Voice, Fax, E-mail, and Alphanumeric pager calls. The only exception is Numeric Pager calls. Typically, you can program a numeric pager destination with only the pager number. **In this case** no additional codes need to be added. However, in instances where the pager service disconnects quickly, or if you must navigate through a voice menu with touch-tones, you will need to add additional codes. To have the Alarm ID number appear on a numeric pager insert the letter "D" into the telephone number. An example is shown below:

Example: 7772233PP6105551234D#

Description of each segment of the beeper number above:

7772233 = pager telephone number
P = 2 second pause
6105551234 = computer telephone number to appear on pager
D = Alarm ID number (4 digits)
# = # (touch-tone)

On the Numeric pager this would appear as "6105551234-0001" which translates to a call from telephone number 610-555-1234 with Alarm ID 0001. To find out exactly what the alarm is, the person would have to call 610-555-1234 and enter the Alarm ID number. The computer would then recite the complete Alarm information and allow the person to enter an acknowledgement code to cancel the dialout process.

ALARM PROGRAMMING

Step Five is to program alarms. Alarm programming for PC Alarm Dialout is identical to programming alarms for normal telephone-line operation. Click the *Alarms* button from the toolbar, or select *Programming*, then *Alarms* from the main menu. This will bring up the Alarms Programming screen. For detailed information on the alarm parameters, see Alarm Programming in chapter 11.

Alarm Acknowledgement

Alarms can be acknowledged when the call is received, by entering the touch-tone acknowledgement code or by calling the computer back and entering the Alarm ID, followed by the acknowledgement code. When the computer answers the call it will prompt you for the Alarm ID number. This must be a 4-digit number. When the number is entered, the computer will recite the SCADA 3000 Unit ID message and then the alarm message. It will then prompt you for the Acknowledgement code for the SCADA 3000 that generated the alarm. Once the alarm is acknowledged, all remaining calls to "Until Acknowledged" destinations for this alarm will be cancelled. Remaining calls to "Inform Only" destinations will continue until all of the messages have been delivered or until the maximum number of calls has been reached.

SAMPLE ALARM DIALOUT SEQUENCE

Voice

When dialing out to a destination programmed as "voice," the software dials the number and immediately starts announcing "SCADA 3000 alarm exists." It will continue repeating this message until the call is answered. Once the call is answered, the software recites the user-recorded identification message for the SCADA 3000 unit with the alarm, then the message identifying the specific alarm that exists. Below is an example of what the software might say during a typical 'voice' dialout:

"SCADA 3000 alarm exists"

"Hello, this is (ID number)"

"(Custom ID message)"

"Alarm Channel 1, (Custom alarm message), is too high. When the alarm occurred, it was 82.90 degrees Fahrenheit."

"Hello, this is (ID number)"

"(Custom ID message)"

"Alarm Channel 1, (Custom Alarm Message), is too high. When the alarm occurred, it was 82.90 degrees Fahrenheit."

"Enter acknowledgment code."

"Enter acknowledgment code."

"Alarm ID 0001. Goodbye."

In this example, the number of Voice Message Repetitions was set to two.

Note: If the call reaches an answering machine, the message will be recorded, but SCADA 3000 will be talking over your outgoing message, so you will probably lose part of the first alarm message repetition. The message will continue and give an alarm ID code. The alarm ID code is needed if you have to call the computer back to acknowledge the alarm. When the computer answers the call it will prompt you for the alarm ID and then ask for acknowledgement. A sample of a call to the computer is shown below.

"SCADA 3000 Alarm exists"

"Enter Alarm ID" (waits for 4-digit entry)

"(Custom ID message)"

"Alarm channel 1, (Custom alarm message), is too high, when the alarm occurred, it was 82.90 degrees Fahrenheit."

"Enter acknowledgement code" (555 acknowledgement code entered)

"Alarm ID 0001 acknowledged, goodbye"

Numeric Pager

When dialing out to a destination programmed as "Pager," the PC Alarm Dialout program leaves the unit ID number (usually the computer's phone number), and the alarm ID number, on the display of a numeric pager. A sample is shown below:

```
610-555-1234-0001
```

To acknowledge the alarm you must call the computer back and enter the alarm ID number and the acknowledgement code. (See voice sample above.)

Alphanumeric Pager

When dialing out to a destination programmed as "Alpha," the PC Alarm Dialout program leaves a text message up to 80 characters long on the display of an alphanumeric pager. A sample is shown below:

```
Water Plant
@610-555-1234
AlarmID:0001 Alarm
Message: Level
TooHigh, 425 inches
```

To acknowledge the alarm you must call the computer back and enter the alarm ID number and the acknowledgement code. (See voice sample above.)

FAX and E-mail

When sending messages via FAX or e-mail, the PC Alarm Dialout program sends a text message. A sample follows.

SCADA3000 Alarm Message

Date: 10/1/00

Time: 8:33:19 AM

To: John Smith
777-2283

From: Water Plant
610-555-1234

Alarm Id: 0003

Alarm Message: Water level too high, 425 inches

To acknowledge the alarm you must call the computer back and enter the alarm ID number and the acknowledgement code. (See voice sample above.)

Chapter 23: DISTRIBUTED CONTROL USING VISUAL BASIC SCRIPT

The SCADA 3000 software allows you to run a control program on the Real-time screen which has the capability of reading and writing to inputs/outputs of all SCADA 3000 units which are actively used on the Real-time screen. All I/O points used in the distributed control program must be referenced on the Real-time screen. The control program can be written in Visual Basic script, Delphi script, or Java script. This chapter will discuss how to write a control program using Visual Basic script. The procedure for writing and running a distributed control program is identical for using the other languages.

A distributed control program is typically used when multiple units are connected via a radio network and they must share information or be controlled from one another. In this scenario all of the SCADA units are online all of the time and I/O values are continuously updated on the Real-time screen. In the case where multiple units are connected via phone lines, the distributed control program will read and write to I/O on an as-connected basis according to the connection frequency programmed on the Real-time screen. In this case, output changes that must be performed on units which are presently unconnected will be queued until a connection is made.

Visual Basic® script is a computer programming language developed by Microsoft® that provides the capability for writing sophisticated control programs. Integrated within the SCADA 3000 Real-time screen editor is a Distributed Control Program editor. This editor allows you to write and compile a program written in the Visual Basic language to perform intelligent input/output control and mathematical functions. Listed within this chapter are explanations on common, available functions along with examples on how to make use of them. Online Help is available from the editor which provides complete documentation for Visual Basic script.

The Distributed Control Program can only run when the Real-time screen is live. The program can be set to run periodically using a programmable delay or continuously by time-slicing the computer's processing capability. A sample template is provided to get you started.

REQUIREMENTS

The Microsoft Visual Basic Scripting host must be installed on your computer. This program can be found on the SCADA 3000 CD.

DISTRIBUTED CONTROL SCRIPT EDITOR

The SCADA 3000 Real-time screen contains a programming utility that allows you to write Distributed Control programs and then run them on your computer. To access the Distributed Control program Editor, click on the *DC* icon from the System Tab of the Real-time screen and then click on the palette of the Real-time screen. The System tab is shown below:

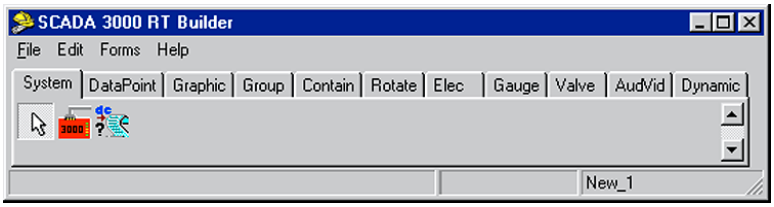

SCADA 3000 RT Builder
File Edit Forms Help
System DataPoint Graphic Group Contain Rotate Elec Gauge Valve AudVid Dynamic
New_1

Figure 1: Real-Time System tab

This will place the Distributed Control management screen (shown below) on the palette.

Figure 2: Distributed Control management screen

To bring up the Distributed Control program editor, click once on the management screen so that it is highlighted, then locate the *EditScript* item on the Property Editor (shown below).

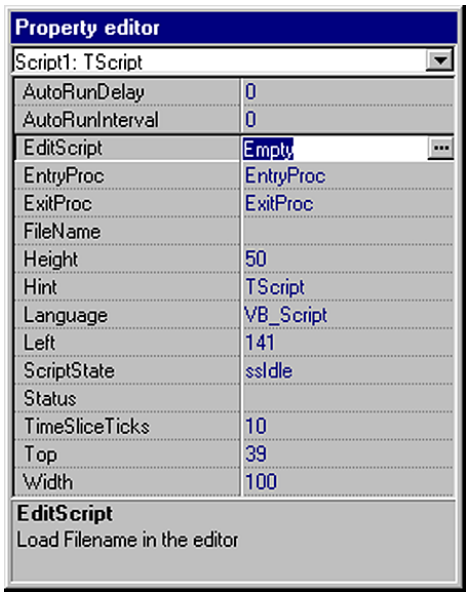


Figure 3: Distributed Control property editor

Now double-click on the filename of the Script program, which should be the word *Empty* initially. This will open the script editor window as shown below:

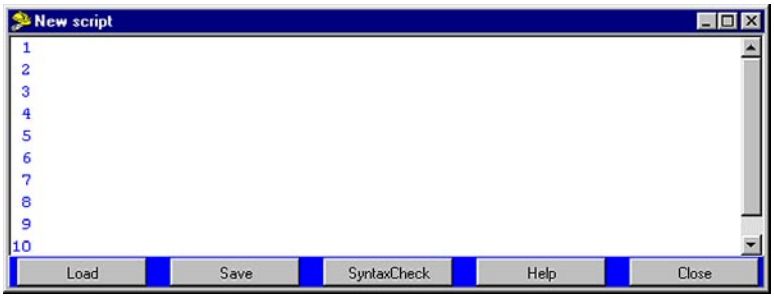


Figure 4: Script Editor window

Use this screen to write and edit your Distributed Control programs. You do not have to be on-line with the SCADA 3000 to write a program. The buttons on the screen are, from left to right: *Load*, *Save*, *SyntaxCheck*, *Help*, and *Close*. The first step in creating a Distributed Control Program is to load the template program (template.vbs). This program includes some pre- and post-program instructions, which are required for your program to run properly. To load the template program, click the *Load* button, then go to the following folder and select the *template.vbs* program: C:\Program Files>SCADA 3000>Samples>Scripts>template.vbs. Click on *template.vbs* and click *OK*. The template program should appear in the window. The program you will write must be entered between the comments labeled "*Begin User Program*" and "*End User Program.*" Do not change or delete any of the instructions that appear before or after the User Program area.

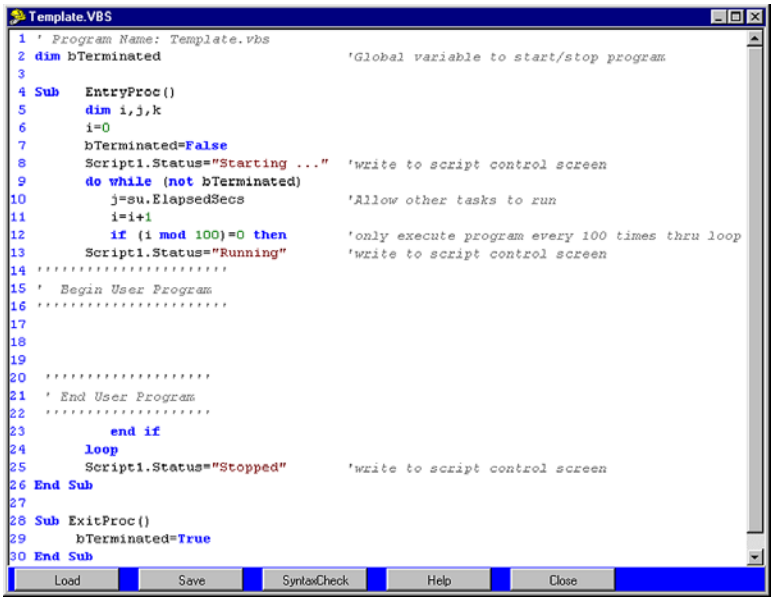


Figure 5: Template program

REFERENCING SCADA 3000 I/O POINTS

Every I/O point or component that is placed on the Real-time screen is assigned a unique tagname which is made up of the *Component Name* and the *Property Name*. The *Component Name* is based on the type of component used on the screen along with a number. Each time that component is used on the screen the number will increase. For example, a DataPointValue component gets the name DataPointValue#. The # part of the name increases by one each time a new point is added to the screen. The *Component Name* can be viewed on the property form when the component is selected. I/O points are referenced in the script program using a combination of the *Component Name* and the *Component Property*. The properties of a component are those listed on the property form. The format for referencing I/O points or components in your script program is *ComponentName.ComponentProperty*.

As an example, if you were going to reference the numerical value of an Input on your Real-time screen you would use the DataPointValue component. The property you would read or write would be the value property. So the complete tagname would be *DataPointValue1.value*. You use the tagnames to read and write to I/O in your Distributed Control program.

Sample Application:

Suppose you have two SCADA 3000 units, one at Location A, the other at Location B. At location A there is a water tank with a level transducer connected to the SCADA 3000. At Location B there is a pump, which is connected to an output on the SCADA 3000. When the pump is turned on at Location B, water will be pumped into the tank at Location A.

A host computer is set up so that it is continuously online with both SCADA 3000 units. A Real-time screen is created, which displays the level in the tank and the On/Off status of the pump. See below.

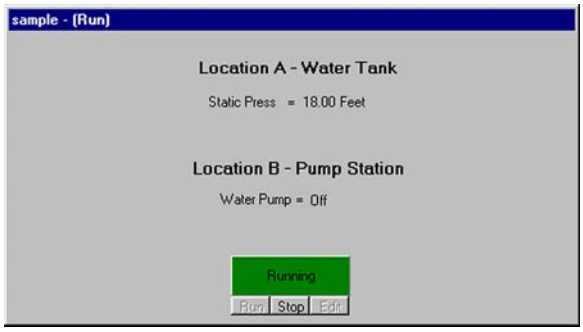


Figure 6: RT Screen

A distributed control program is running which will turn on the pump if the level falls below 10 feet. The pump will be turned off when the level reaches 35 feet. A listing is shown below:

```
''''''''''''''''''''''''
'  Begin User Program
''''''''''''''''''''''''


' In the following program, DataPointValue1 is the tanklevel and
' DataPointValue2 is the Pump On/Off status. The program will turn on
' the pump if the level drops below 10 feet. The pump will stay on
' until the level is above 35 feet. A static label is used to display
' the On/Off status of the pump on the Realtime screen.

dim tanklevel,pumpcontrol             'Declare variables
    tanklevel=DataPointValue1.Value  'Assign datapoint component value from -
                                      'realtime screen to tanklevel variable

if tanklevel<10 then                  'Perform logic
   pumpcontrol=1                      '     .
end if                                '     .
if tanklevel>35 then                  '     .
   pumpcontrol=0                      '     .
end if

DataPointValue2.Value=pumpcontrol     'Assign pumpcontrol value to datapoint -
                                      'component on realtime screen
if DataPointValue2.Value=0 then       'Check if pump is on or off and print -
   Label5.Caption="Off"               'status on realtime screen
else Label5.Caption="On"
end if
''''''''''''''''''''''''
' End User Program
''''''''''''''''''''''''
```

Syntax Checking

Once you have finished writing your Distributed Control program you must check it for syntax errors. Simply click the *SyntaxCheck* button. The editor will provide a message to indicate if there were any errors detected, along with the location of the error.

Saving your Program

When you have finished editing your program be sure to save it by clicking the *Save* button. Enter a unique name for your program and click *OK*.

> **Note**: Do not use *template.vbs* as a filename as this will overwrite the original template program. Click *Close* to exit the editor.

Running your Program

After the program passes the Syntax Check you can try to run it. Remember that the program will actually read and control I/O, so it's highly recommended that you **test** the program in a scenario where mistakes in logic won't damage equipment or place people in danger. To run your program, the Real-time screen must first be running. The reason for this is that the I/O values used in the program are read from and written to the Real-time screen. The Real-time screen handles all data communications between the SCADA units and the Distributed Control program. Once the Real-time screen is running, click the *Run* button on the Distributed Control management screen. A message should appear that first says *Starting...*, then *Running*. To stop the program click *Stop*.

PROPERTIES OF THE SCRIPT COMPONENT

If you click once on the Distributed Control management screen, the property editor will display the component properties.

Listed below are descriptions of each property of the script component. Note that most of the properties listed below will never have to be changed.

AutoRunDelay - Allows you to have the script program start automatically when the realtime screen starts. Enter a number in units of seconds. A value of 0 disables the feature.

AutoRunInterval - Allows the script to restart automatically after a time period has expired. Enter the time in seconds. A value of 0 disables this feature.

EditScript - Shows the filename of the associated script program. Double-clicking the filename will bring up the script editor.

EntryProc - Name of the entry procedure for the script program (DO NOT CHANGE).

ExitProc - Name of the exit procedure for the script program (DO NOT CHANGE).

FileName -Path and filename of the associated script program.

Height - Height of the script control screen.

Hint - Text that appears when you move the cursor over the Distributed Control management screen.

Language - Selection of development language. The choices are Visual Basic script, Delphi script or Java script.

Left - Position of the left side of the script control screen.

ScriptState - Allows you to change the state (run/stop/idle) of the script program.

Status - Displays the current status of the script program.

TimeSliceTicks - Time (in milliseconds) the script is yielded during timer calls.

Top - Position of the top-side of the script control screen

Width - Width of the script control screen

ABOUT THE VISUAL BASIC LANGUAGE

Visual Basic® is a structured programming language that is similar to other structured languages like C. The **structure** of Visual Basic works as if you ask a question or pose a problem, and then go through the simple steps of information gathering and action to solve the problem. For example, if your alarm clock is ringing and you want it to stop ringing, you must hit the OFF button to turn it off, otherwise nothing will happen. If you want the Distributed Control program to turn off the alarm clock, you must first tell it how to recognize that the alarm is ringing, then give it the command to turn the alarm off when it recognizes that it is ringing, otherwise do nothing. To give the Distributed Control program this information, you must translate English commands into Visual Basic commands.

The Visual Basic language structure is fairly simple. Each program must have a beginning and an end. Each individual statement within the program must open, provide its information, and then close. Think of it as writing sentences—without the proper sentence structure and punctuation, you will get syntax errors, and the computer will not be able to receive the message or understand what to do. **There is much more information and detail on the Visual Basic script language in the online help.**

The Distributed Control program comprises the following elements:

STRUCTURE: To write a valid Visual Basic program, regardless of length, you must follow its simple structural elements.

KEYWORDS: These are short words or symbols that define variables or execute specific actions within a Visual Basic program. They include operators, assignment, comparators, and comments.

PREDEFINED VARIABLES: These are variables that have specific predefined values that are automatically updated and cannot be changed by the user.

FUNCTIONS: These are word commands that perform a predefined function within a Visual Basic program.

ARRAYS: This is a more advanced programming type that significantly shortens and simplifies long programs.

ERROR HANDLING: When a program is compiled, it is scanned for language related errors. An error message, including the type of error, is displayed.

EDITING TOOL: The SCADA 3000 Realtime Editor provides a Distributed Control program editing utility that allows you to write, compile, and run Distributed Control programs. This chapter provides basic instruction in the Visual Basic language and gives all the commands that are valid for use with SCADA 3000. Sample programs are included.

STRUCTURE

Listed below is information that will help demonstrate the Visual Basic language structure as it is used in the SCADA 3000. Refer to the following pages for explanation of the keywords, functions, and commands used within the sample programs.

All programs must start with the predefined template (template.vbs). Your specific program must be inserted between the comments labeled *Begin User Program* and *End User Program.*

```
' Begin User Program
' End User Program
```

*__Note__: All SCADA 3000 Distributed Control Programs must include some pre-program and post-program instructions. These have been pre-written for you and

are included in the distributed control template program (template.vbs) located in the SCADA 3000>Samples>Scripts folder.

KEYWORDS

The following is a partial list of valid components of the Visual Basic language that can be used within the SCADA 3000 Distributed Control program.

Dim

If . . . Then . . . Else

For . . . Next

Do . . . Loop

DEFINITIONS

Dim Statement

Declares variables and allocates storage space.

Usage

```
Dim varname[([subscripts])][, varname[([subscripts])]] . . .
```

Arguments

varname

Name of the variable; follows standard variable naming conventions.

subscripts

Dimensions of an array variable; up to 60 multiple dimensions may be declared. The subscripts argument uses the following syntax:

upperbound [,upperbound] . . .

The lower bound of an array is always zero.

variable name definitions

A named storage location that can contain data that can be modified during program execution. Each variable has a name that uniquely identifies it within its level of scope.

Variable names:

- Must begin with an alphabetic character.
- Can't contain an embedded period or type-declaration character.
- Must be unique within the same scope.
- Must be no longer than 255 characters.

If...Then...Else Statement

Conditionally executes a group of statements, depending on the value of an expression.

```
If condition Then statements [Else elsestatements ]
```

Or, you can use the block form syntax:

```
If condition Then
   [statements]
[ElseIf condition-n Then
   [elseifstatements]] . . .
[Else
   [elsestatements]]
End If
```

Arguments

condition

One or more of the following two types of expressions:

A numeric or string expression that evaluates to **True** or **False**. If *condition* is Null, *condition* is treated as **False**.

An expression of the form **TypeOf** *objectname* **Is** *objecttype*. The *objectname* is any object reference and *objecttype* is any valid object type. The expression is **True** if *objectname* is of the object type specified by *objecttype*; otherwise it is **False**.

statements

One or more statements separated by colons; executed if *condition* is **True**.

condition-n

Same as *condition*.

elseifstatements

One or more statements executed if the associated *condition-n* is **True**.

elsestatements

One or more statements executed if no previous *condition* or *condition-n* expression is **True**.

For...Next Statement

Repeats a group of statements a specified number of times.

```
For counter = start To end [Step step]
  [statements]
Next
```

Arguments

counter

Numeric variable used as a loop counter. The variable can't be an array element or an element of a user-defined type.

start

Initial value of *counter*.

end

Final value of *counter*.

step

Amount *counter* is changed each time through the loop. If not specified, *step* defaults to one.

statements

One or more statements between For and Next that are executed the specified number of times.

Do...Loop Statement

Repeats a block of statements while a condition is True or until a condition becomes True.

```
Do     [{While | Until} condition]
       [statements]
       [Exit Do]
       [statements]
Loop
```

Or, you can use this syntax:

```
Do
      [statements]
      [Exit Do]
      [statements]
Loop [{While | Until} condition]
```

Arguments

condition

Numeric or string expression that is **True** or **False**. If *condition* is Null, *condition* is treated as **False**.

statements

One or more statements that are repeated while or until *condition* is **True**.

OPERATORS

Symbols used to execute mathematical operations and determine whether a particular condition exists.

+	Adds two values
-	Subtracts two values
*	Multiplies two values
/	Divides two values and returns a floating point result
\	Divides two values and provides an integer result
Mod	Finds the remainder of a division
^	Raises a number to the power of an exponent
Or	Checks if one condition or another exists
And	Checks if one condition and another exists

ASSIGNMENT -Symbol used to assign a numeric value.

=	assigns a new value to a variable

COMPARATORS -Symbols used to compare numeric values to each other

<	Less than
>	Greater than
=	Is equal to
<>	Is not equal to
<=	Less than or equal to
>=	Greater than or equal to

COMMENTS -Text used to add notes to a program. Typically used to briefly explain what the program or statement is doing. They may be used freely to make a program easier to understand. Comments are ignored by the program. A comment is identified by a single quote at the beginning of the line.

' comment

MONTH, DAY, YEAR, HOURS, MINUTES, SECONDS, & WEEKDAY

These variables represent the values from the real-time clock in your PC. They do not have to be declared in a Dim statement. The variable names are as follows:

```
Month(Now)      ' Holds the present value from 1 to 12.
Day(Now)        ' Holds the present value from 1 to 31.
Year(Now)       ' Holds the present value in 4 digit format (e.g. 2001).
Hour(Now)       ' Holds the present value in 24 hour time from 0 to 23.
Minute(Now)     ' Holds the present value from 0 to 59.
Second(Now)     ' Holds the present value from 0 to 59.
Weekday(Now)    ' Holds the present value from 1(Sunday) to 7(Saturday).
```

Example: This program will set DataPointValue4 to 35 at midnight.

```
if (Hour(Now)=0) and (Minute(Now)=0) and (Second(Now)=0) then
    DataPointValue4.Value = 35
end if
```

PROGRAMMING METHODOLOGY

There are three methods that will typically be used when creating a control system based on a distributed control program. The method you choose will depend on your application, how many units you have, and other factors which may be specific to your application. The three methods are described below:

Method 1

Your distributed control program performs all control logic. No logic programs run on the SCADA units themselves.

Advantages

- This allows for centralized control and there is only one control program to create and maintain.

Disadvantages

- You will need all I/O points required for running your logic program to be on the realtime screen.
- If the PC crashes or the program must be halted, then all control stops.

Method 2

Your distributed control program only passes I/O information from one unit to another. All control logic is performed in the SCADA units themselves using a C or Ladder program. Information can easily be read from unit A and written into the C or ladder variables of unit B.

Advantages

- Control is performed locally at the SCADA unit. If the PC running the distributed control program crashes or if the program must be halted, local control will continue to function.
- This minimizes the number of I/O points on your realtime screen.

Disadvantages

- You will need to have separate control programs running for each SCADA unit which may be more complicated to write and maintain.

Method 3

A combination of Methods 1 & 2 where both the distributed control program runs control logic and each SCADA unit also runs its own specific control program.

Advantages

- Sophisticated control can be attained along with localized control.
- If the PC crashes or the distributed control program must be halted, localized control can still be performed.

Disadvantages

- Multiple control programs must be written for both the host PC and each individual SCADA unit.
- Program maintenance could be difficult.

SAMPLE PROGRAM

The following program illustrates simple passing of I/O values from one SCADA unit to another.

```
''''''''''''''''''''''''
'  Begin User Program
''''''''''''''''''''''''
' Sample program which assumes that a Tank is located at site A and a
' Pump is located at site B.  The program passes the tanklevel from
' unit A to unit B where it is stored in a ladder variable.  This
' allows unit B to control the pump.  The On/Off status of the pump is
' then sent back to unit A where it is also stored in a ladder
' variable.
' Definitions:
'    DataPointValue1 is the TankLevel at unit A
'    DataPointValue2 is the Pump Status at unit B
'    DataPointValue3 is a Ladder Variable in unit A
'    DataPointValue4 is a Ladder Varaiable in unit B

DataPointValue4.Value=DataPointValue1.Value  'Assigns DataPointValue1
                                             'to DataPointValue4

DataPointValue3.Value=DataPointValue2.Value  'Assigns DataPointValue2
                                             'to DataPointValue3

''''''''''''''''''''
' End User Program
''''''''''''''''''''
```

Appendix A: CHECKING YOUR SCADA 3000 FOR PROPER OPERATION

We recommend that you test your Sensaphone weekly to be sure it is functioning properly. This will ensure that when a problem arises the Sensaphone will be ready to alert the appropriate personnel.

There are several tests that can be performed:

1) Call the unit and listen to the Status Report. This will test the unit's ability to answer the phone and speak a message. It will also verify that the inputs are reading properly, the alarm conditions are OK, the electricity is on, the control programs are running, and that the battery is OK.

2) Create a test alarm on an input and allow the unit to contact all programmed telephone numbers. This will make sure that the Sensaphone is programmed properly. It will also prepare personnel to respond appropriately when they receive a call from the Sensaphone.

3) Test the battery (if connected) by unplugging the external power supply and making sure that the Sensaphone continues to function. Check the battery voltage with a DC voltmeter or through the SCADA keypad in the Diagnostics menu, to make sure it is at least 1 3V. Reconnect the power supply when finished.

4) Test the internal modem by calling the Sensaphone from a computer and logging on. While connected, run the motherboard diagnostics to verify that all internal system parameters are correct.

Appendix B: SCADA 3000 SPECIFICATIONS

Number of Inputs	16
Input Types	Normally Open/Normally Closed Dry Contact, Run Time, 0-5V Analog Voltage, 4-20mA Current Loop, 10K Thermistor
Input Voltage Range Vin+ to Vin-	0 - 5V
Maximum Voltage on Vin+ or Vin-	+12V (4-20mA & 0-5V configuration only)
Minimum Voltage on Vin+ or Vin-	-12V (4-20mA & 0-5V configuration only)
10K Thermistor Temperature Range	-100° to 300° F (-65° to 150° C)
Input Resolution	12 bits (1.22mV/count)
Typical A/D Converter error	±1 LSB
Input Protection	Redundant, fast-acting, low-leakage, over/under voltage diode clamps
Dry contact/thermistor sense voltage	5V through 10KΩ
4-20mA load resistance	237 Ohms .1% 1/4 Watt
0-5V input impedance	10^{12} Ω
Number of Latching Relay Outputs	8
Rated Load of Relay (resistive)	2A 125VAC, 2A 30VDC
Rated Load of Relay (inductive)	1A 125VAC, 1A 30VDC
Max. Relay Operating Voltage	125VAC, 125VDC
Max. Relay Switching Capacity (resistive)	250VA, 60W
Max. Relay Switching Capacity (inductive)	125VA, 30W
Minimum permissible load on Relay	10mA, 5VDC
Relay Contact Protection	275V Metal-Oxide-Varistor on each output
Input/Output Expansion	Up to 15 additional I/O modules
Network Data Rate	153.6 Kbps
Bus Termination Impedance	120 Ohms
Number of Serial Ports	Two: (1) DB9 DTE and (1) DB9 DCE
Baud Rate (Bps)	Selectable: 1200, 2400,4800, 9600, 19.2K, 38.4K
Liquid Crystal Display	4 rows by 20 Characters with LED Backlight
LED Indicators	Alarm, Fault, Control, Monitoring, Battery, Phone & Outputs 0-7
Options	Voice Module, part #FGD-3500 33.6K Modem, part #FGD-3400

Power Supply Requirements **Input Voltage (without battery)** **Input Voltage (with battery)** **Current consumption** **Note: Options & I/O modules will increase power consumption.**	 10-15VDC 14.7-15VDC 215mA (min) 270mA w/backlight on add 8mA w/voice option add 85mA w/modem option add 38mA per 4-20mA loop powered from 24V supply
Aux Power Output (for I/O modules)	10-14VDC
24V Output Current	320 mA max.
Battery Back-up (Optional)	12V Lead-acid Gel-cell 5.0AH to 18AH Automatic charge control & deep discharge protection
Program Memory Back-up Battery	CR2430 Lithium Battery
Power Fuse Rating & type	4A 250V, Size TR-5 (Wickmann #3721400041)
Terminal Block Wire Size	#14AWG to #24AWG
Operating temperature	0 to 70 degrees Celsius (32 to 158 degrees F)
Storage Temperature	-20 to 70 degrees Celsius (-4 to 158 degrees F)
Humidity	0 to 90%, non-condensing
Dimensions	9.4" x 12.2" x 2.0"
Weight	3.0 lbs.
Enclosure	Aluminum Housing with integral mounting flanges for wall or panel installation.

Appendix C: LIST OF SYSTEM EVENTS

A LIST OF EVENTS THAT WILL BE STORED IN THE EVENT LOGGER.

Parameter:

Start up completed

Diagnostic *(0 — hardware reset*
1 — full reset)

Data logger *(0 — reset*
1 — start
2 — stop
3 — download
4 — full)

Status password successful

Programming password successful

Password attempt failed

Alarm detected *(Alarm Number)*

Alarm not configured

Alarm acknowledged by max calling rounds

Alarm acknowledged by 15 calls to 1 destination

Alarm acknowledged by inbound phone call

Acknowledgement received

No acknowledgement received on outbound phone call

Dialing "until Acknowledge" destination *(Destination number)*

Dialing "Inform only" destination *(Destination number)*

Phone busy

No answer

Voice call answered

Modem Connect *(Speed)*

Fax connect *(Speed)*

Email connect *(Speed)*

Numeric pager connect

Alpha numeric pager connect *(Speed)*

Message delivered

Hangup

Inbound phone call received

Giving voice report

Inbound phone call terminated

Alarm detected after reset time expired *(Alarm number)*

Voice report given

Dialing report destination *(Destination number)*

Alarm cleared *(Alarm number)*

No dial tone

Alarm low detected *(Alarm number)*

Alarm high detected *(Alarm number)*

Invalid hardware configuration for dial out

Fault condition exists *(Fault type)*

Control Switch Enabled

Control Switch Disabled

Monitor Switch Enabled

Monitor Switch Disabled

Anonymous Login

Appendix D: 10k THERMISTOR CURVE DATA

DEGREES (Celsius)	DEGREES (Fahrenheit)	RESISTANCE (Ohms)
-37	-35	203.60K
-35	-30	173.60K
-32	-25	148.30K
-29	-20	127.10K
-26	-15	109.20K
-23	-10	94.07K
-21	-5	81.23K
-18	0	70.32K
-15	5	61.02K
-12	10	53.07K
-9	15	46.27K
-6	20	40.42K
-4	25	35.39K
-1	30	31.06K
2	35	27.31K
4	40	24.06K
7	45	21.24K
10	50	18.79K
13	55	16.65K
16	60	14.78K
18	65	13.15K
21	70	11.72K
24	75	10.46K
27	80	9.35K
30	85	8.38K
32	90	7.52K
35	95	6.75K
38	100	6.08K
41	105	5.48K
44	110	4.95K
47	115	4.47K
49	120	4.05K
52	125	3.67K
55	130	3.33K
58	135	3.31K
60	140	2.76K
63	145	2.52K
66	150	2.30K
69	155	2.10K
71	160	1.92K
74	165	1.76K
77	170	1.61K
80	175	1.48K
83	180	1.36K
86	185	1.25K
88	190	1.16K
91	195	1.07K
94	200	0.98K
97	205	0.91K

Appendix E: COMMUNICATIONS PORTS

RS232 COM PORTS

SCADA 3000 has two RS232 ports, one configured as DCE 9-pin female (COM 1) and the other as DTE 9-pin male (COM 2). COM 1 is designed to connect directly to a personal computer for local communications/programming. COM 2 is used to communicate with SCADA 3000 via a radio modem (optional). The radio modem would connect to COM 2 directly with the appropriate cable. The baud rate and communication type (radio or standard) for each port is selectable via jumpers located under the top cover (see Figures below). The default Baud rates and communication types are listed below.

COM 1 Default Setting: 38,400 bps, Standard

COM 2 Default Setting: 38,400 bps, Standard

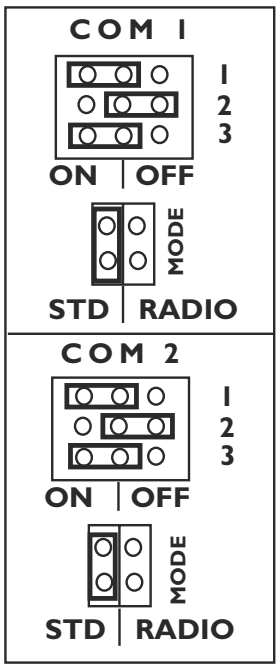
BAUD RATE (bps)	JUMPER SETTINGS 1	2	3
1200	off	off	off
2400	on	off	off
4800	off	on	off
9600	on	on	off
19,200	off	off	on
38,400	on	off	on
57,600 (COM 1 only)	off	on	on

Figure 1: Baud Rate Selection Jumpers

Note: The Radio setting activates the RTS, CTS and DCD pins on the connector specifically for radio communications. Do not use this setting for normal PC communications.

Pin Description and Data Direction

The pin descriptions and signal direction for each COM port follow:



Example: 38,400

Example: 38,400

Figure 3: Com Ports

COM 1 (DCE)

Pin	Description	Direction
1 DCD	Data Carrier Detect	Output
2 RD	Receive Data	Output
3 TD	Transmit Data	Input
4 DTR	Data Terminal Ready	Input
5 SG	Signal Ground	
6 DSR	Data Set Ready	Output
7 RTS	Request to Send	Input
8 CTS	Clear to Send	Output
9 N.C.	No Connection	

COM 2 (DTE)

Pin	Description	Direction
1 DCD	Data Carrier Detect	Input
2 RD	Receive Data	Input
3 TD	Transmit Data	Output
4 DTR	Data Terminal Ready	Output
5 SG	Signal Ground	N/A
6 DSR	Data Set Ready	Input
7 RTS	Request to Send	Output
8 CTS	Clear to Send	Input
9 N.C.		No Connection

Appendix F: UPDATING SCADA 3000 FIRMWARE

The firmware in your SCADA 3000 can be upgraded without having to send the unit back or replace any chips. This is because of an advanced technology called Flash memory. Inside the SCADA 3000 are two Flash memory chips that contain the firmware for each of the units microprocessors. One processor is dedicated to Communications, the other to Control. The firmware is stored in a format called a Hex file. When upgrading the firmware with new Hex files it is recommended that you upgrade both Flash memory chips.

> **Warning: Be sure to save all of you programming and to download any information in the data logger and event log. The memory in the SCADA 3000 will be completely reset after performing the upgrade.**

The first step to upgrading the firmware is to retrieve the Hex files. The files are available three ways:

1) **Download** the Hex files from the internet using the Sensaphone Software Internet Upgrade Program (see Appendix F).

2) Have them **E-mailed** from our tech support personnel (Contact support@sensaphone.com)

3) Have the files mailed to you on a **floppy disk**. (Call (610)558-2700)

Copy the two Hex files to your hard drive in the SCADA 3000 folder (C:\Program Files\SCADA 3000\Firmware).

Once you have the files and have saved the data in your SCADA 3000, you may upgrade the firmware via an RS232 connection or via a modem connection.

Upgrading via RS232

Go online with the unit. From the Diagnostics menu select Flash Upgrade. The following form will appear:

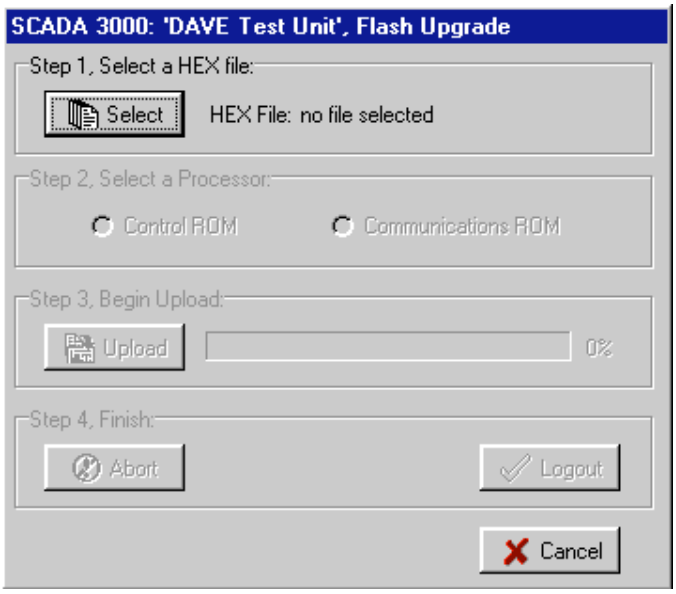


Figure 1: Flash Upgrade form

Instructions:

1) Click on the *Select* HEX file button.

2) Open the SCADA 3000 firmware folder.

3) The two HEX files should appear. Click on the *Control* HEX file and click *Open.*

4) In the *Select a Processor* field, click "Control ROM."

5) Click the begin *Upload* button. After a few seconds the upgrade will begin. When its 100% finished, click on the *Loqout* button. During the Upload process, the LCD will display "Uploading Control Flash Programming." After the upload process, the display will change to "Reprogramming Control Flash Memory." This will take about one minute, after which the display will return to its normal scrolling messages. Do not disrupt the unit until the reprogramming message has disappeared.

6) Go online with the unit again.

7) Click on Diagnostics>Flash Upgrade, from the main menu again.

8) Click on the *Select* HEX file button.

9) Open the SCADA 3000 firmware folder.

10) The two HEX files should appear. Click on the *Comm* HEX file (communications) and click *Open*.

11) In the *Select a Processor* field, click "Communications ROM."

12) Click the begin *Upload* button. After a few seconds the upgrade will begin. When its 100% finished, click on the *Loqout* button. During the Upload process, the LCD will display "Uploading Communications Flash Programming." After the upload process, the display will change to "Reprogramming Communications Flash Memory." This will take about one minute, after which the display will return to its normal scrolling messages. Do not disrupt the unit until the reprogramming message has disappeared.

The Flash upgrade is now complete. You may go back on line and restore your programming.

Appendix G: SOFTWARE INTERNET UPGRADE

The Internet Upgrade program allows you to retrieve SCADA 3000 Software and Flash Firmware updates via the internet. The program will check to see what software versions you have and then check our website and list the files that need to be updated. Follow the instructions below to configure the program and update your files.

> **Note**: You must have internet access to use this program. You should connect to the internet before running the upgrade.

STARTING THE PROGRAM

Click on the Internet Upgrade program Icon from the desktop (or select it from the Windows Start menu).

The program will display all pertinent SCADA files as shown below:

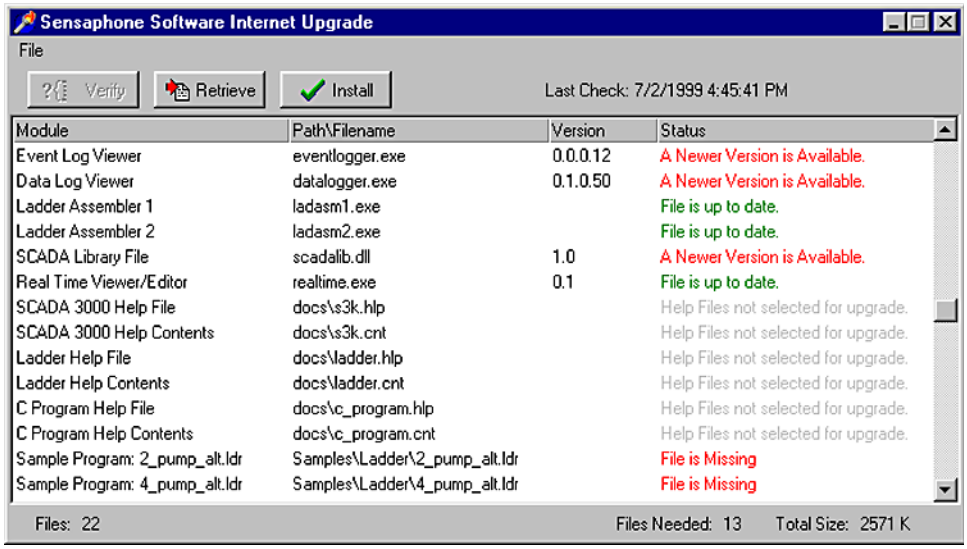

Sensaphone Software Internet Upgrade
File
Verify
Retrieve
Install
Last Check: 7/2/1999 4:45:41 PM
Module | Path\Filename | Version | Status
Event Log Viewer | eventlogger.exe | 0.0.0.12 | A Newer Version is Available.
Data Log Viewer | datalogger.exe | 0.1.0.50 | A Newer Version is Available.
Ladder Assembler 1 | ladasm1.exe | | File is up to date.
Ladder Assembler 2 | ladasm2.exe | | File is up to date.
SCADA Library File | scadalib.dll | 1.0 | A Newer Version is Available.
Real Time Viewer/Editor | realtime.exe | 0.1 | File is up to date.
SCADA 3000 Help File | docs\s3k.hlp | | Help Files not selected for upgrade.
SCADA 3000 Help Contents | docs\s3k.cnt | | Help Files not selected for upgrade.
Ladder Help File | docs\ladder.hlp | | Help Files not selected for upgrade.
Ladder Help Contents | docs\ladder.cnt | | Help Files not selected for upgrade.
C Program Help File | docs\c_program.hlp | | Help Files not selected for upgrade.
C Program Help Contents | docs\c_program.cnt | | Help Files not selected for upgrade.
Sample Program: 2_pump_alt.ldr | Samples\Ladder\2_pump_alt.ldr | | File is Missing
Sample Program: 4_pump_alt.ldr | Samples\Ladder\4_pump_alt.ldr | | File is Missing
Files: 22
Files Needed: 13
Total Size: 2571 K

Figure 1: Internet Upgrade Main Window

With the default settings in place you may verify, retrieve and install the software upgrade as follows. If you wish to change the default settings, select *Options* from the *File* pulldown menu.

Verify

Click this button to have the Upgrader program log in to the Phonetics server and compare the software version at Phonetics with the software version on your computer. The Status list will be updated to reflect whether your files are up to date, or *if a newer version is available*. If a newer version is available and you would like to upgrade now, click the *Retrieve* button.

Retrieve

Click this button to download the newest software versions from the Phonetics server. They will be saved to a folder on your computer called NewVer.

Install

Click this button to install the files in the new version directory.

OPTIONS

From the *File* menu, select *Options*. The following form with 4 tabs should appear.

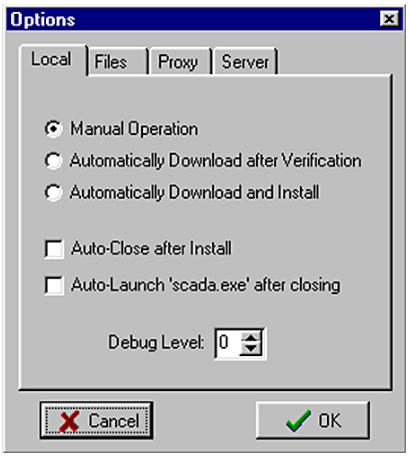


Figure 2: Options, Local tab

Local Tab

This screen provides several options which allow the upgrade program to automatically execute certain functions. A description of each is listed below.

Manual Operation: Requires the user to manually verify which files need to be upgraded, click the *Retrieve* button to download the files, and separately click the *Install* button to complete the upgrade.

Automatically Download after Verification. Clicking on the Start button will verify and download automatically, but will not install.

Automatically Download and Install. In this mode, clicking on the one Start button verifies, downloads and installs.

Auto-Close after Install will close the upgrade program once the update has been installed.

Auto-Launch will open your SCADA software immediately after closing.

Note: If you check the Automatically Download and Install, the Auto-Close after Install, and the Auto-Launch buttons, your upgrade will be installed automatically, the install program will close, and your SCADA program with updated software will launch.

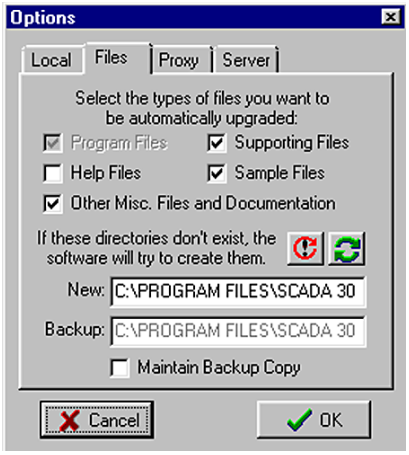


Figure 3: Options, Files tab

File Tab

Click in the boxes to select the type of files you want automatically upgraded. Selecting fewer file types will speed the download process over slow connections.

Program Files are all required files.

Help Files consist of any of the Winhelp files accompanying your software.

Other Misc. Files and Documentation include manuals, specs, and so forth. These will also include bitmap graphics to assist you in building Real-Time screens.

Supporting Files are Firmware upgrades.

Sample Files are the sample C, Ladder, and Real-Time screen files.

The two buttons to the right are resets. The **Reset to Default** button will reset the directory names where the newly downloaded files and backup files will be saved, as shown in the *New* and *Backup* fields. The **Reset to Previous** button can be used to revert back to the previous directory names where newly downloaded files and backup files will be saved.

New: Shows the drive and directory name where downloaded files will be saved.

Backup: Shows the drive and directory name where a backup copy of the old SCADA 3000 files will be saved if "Maintain Backup Copy" has been selected.

Maintain Backup Copy, if checked, creates a backup of the previous version of software. A "Revert to Backup" option in the file menu will be created if you choose this. The default setting is Off.

Figure 4: Options, Proxy tab

Proxy Tab

If you have to go through a proxy server in order to access the internet, check the "Need to use Proxy Server" box and type in the settings you use. If you are unsure, see your System Administrator.

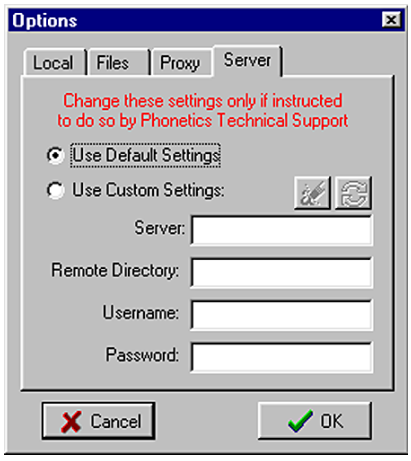


Figure 5: Options, Server tab

Server Tab

The default settings should be maintained on this tab. The boxes will be blank in the default setting. Before entering any custom settings, you should contact Phonetics Technical Support either by phone at (610)558-2700 or email at support@sensaphone.com

The two buttons are resets. The lefthand one clears any settings you've entered in the boxes below. The second, to the right, resets those settings to their previous values.

Appendix H: TROUBLESHOOTING

Why does my SCADA 3000 begin dialing as soon as I turn it on?

• SCADA 3000 will dial out for only two reasons, to deliver an alarm message or a report. If the alarm LED is blinking, an unacknowledged alarm exists and SCADA 3000 is trying to deliver an alarm message. If the alarm LED is either off or on steady, the unit is dialing out to deliver a report.

• The shunt for setting your input type may be in the wrong position, causing a false alarm condition. Check that it is in the correct position for the type of sensor you're using.

• The unit might have an unacknowledged alarm for which it's still trying to dial out and deliver its message. Even though the condition no longer exists, the unit will attempt to deliver its message until it is finally acknowledged. If this is the case, either go on-line with a PC and clear the alarm; or call the unit back and give the Touch-Tone acknowledgment code. You can also plug a telephone into the Voice Record jack and give the Touch-Tone acknowledgment code.

On site, you can also check and acknowledge an alarm via the Keypad.

1) Press the MENU button, arrow-down to the Alarms category, and press ENTER.
2) Press the down-arrow key until you reach the alarm you need to acknowledge. Press ENTER.
3) Press the down-arrow key until the LCD displays *Acknowledge Alarm*. Press ENTER and the alarm will be acknowledged. If you continue to press the down-arrow key, the *Acknowledge All* option will be displayed. If you press ENTER while this message is showing, all alarms will be acknowledged.

Why can't I connect with my SCADA 3000?

Having trouble connecting locally

Some common causes:

• SCADA 3000 is connected to your PC's parallel port (LPT1) instead of a serial COM port (COM1, COM2, etc.).

• SCADA 3000 is connected to the wrong serial COM port. Most PC's have multiple serial ports. Make sure the port connected to the SCADA 3000 unit is the same port shown the UNIT DATABASE form.

• Check that the SCADA 3000 software has configured the COM port as a direct Serial Port connection. Click Comm>Setup from the Sensaphone Control Center Main Menu. Click the appropriate Comm port tab and check that the port is Enabled, the Type is set to Serial Port, and the Port Baud Rate is set to 38400.

Having trouble connecting using a modem

<u>If your modem won't dial at all:</u>

• Make sure that your modem is connected to a serial COM port on your PC, not a parallel (LPT) port.

• Check that the PC's COM port connected to the modem is the same port shown on the UNIT DATABASE form.

• Check that the SCADA 3000 software has configured the COM port to connect to a Phone

Modem. Click Comm>Setup from the Sensaphone Control Center Main Menu. Click the appropriate Comm port tab and check that the port is Enabled and the Type is set to Phone Modem.

• Make sure that your modem is turned on and plugged into a working telephone line.

If your modem dials the SCADA 3000, but won't connect:

•Rings Until Answer may be too high. If Rings Until Answer is set too high, the SCADA 3000 software will time out before the 3000 can answer the call. Increase Dial Time Out or reduce the number of Rings Until Answer.

• Carrier Wait Time may be too short.

• Dial Time Out may be too short.

• COM port Baud Rate setting may be too fast. This is frequently the case when using a cellular telephone to make a modem connection to a SCADA 3000. Wireless phones are more subject to noise and interference than land lines so you may need to slow down the baud rate to establish and maintain a reliable connection.

1) Click Comm>Setup from the Sensaphone Control Center Main Menu.

2) Click the appropriate Comm port tab and select a slower Port Baud Rate until a reliable connection is established.

• Check your Modem Setup String. SCADA 3000 software allows you to enter a setup string of modem AT commands. These are usually only needed if you are having trouble getting your modem to connect to a SCADA 3000 unit. See your modem Operators Manual for a list of AT commands supported by your modem.

If your modem connects with a SCADA 3000, but the connection sometimes drops out:

1) Click Comm>Setup from the Sensaphone Control Center Main Menu.

2) Click the appropriate Comm port tab and try increasing the values of the following:

• Check/Change Packet Retries. Information flows back and forth between SCADA 3000 and your PC in data 'packets'. Both units automatically check for packet transmission errors. When a bad packet is detected, it gets re-sent. Packet Retries determines how many times a packet is re-sent before a communication error is logged.

• Check/Change Retry Delay. Once a packet is sent, this is how long SCADA 3000 software will wait for a response from the 3000 unit before re-sending it.

• Check/Change Packet Time Out. This is the amount of time between the bytes of data that a make up a packet (also known as Inter-character Delay Time).

Will SCADA 3000 retain its programming if there's been a long power failure and the backup battery dies?

When all power is removed from the SCADA 3000, voice messages and unit programming are retained in nonvolatile memory by an internal 3V lithium coin-cell battery. This battery should provide at least 2 years of data storage without power and up to 10 years of intermittent use.

I've set up a Polling schedule, but my PC won't make polling calls to my SCADA 3000 units?

In order for your PC to actually make the scheduled Polling phone calls to units, you must set up a polling schedule for each unit and enable the master polling control from the polling setup

form. (See the Polling Section of the "Phone Programming" chapter for instructions.)

My SCADA 3000 unit got damaged and had to be replaced. Can I use the original unit's saved programming to quickly program the replacement unit?

Yes. To quickly program the replacement SCADA 3000, do the following:

1) First, make sure all the correct hardware connections and settings have been made. This includes wiring up inputs (including jumper shunt settings), outputs, communication bus, power supply, backup battery, etc.

2) Click the RTU's button on the Sensaphone Control Center toolbar. The UNIT DATABASE form will appear.

3) Choose the original SCADA 3000 unit from the text box.

4) Click 'Online'.

5) Once a connection has been established, the SCADA 3000 Unit Name Error form pops up. Since you are loading saved programming into a new unit, select "Use Name in Database:_____". This copies the unit name from the saved file to the new SCADA 3000 unit.

6) From the SCADA 3000 RTU form, (NOT the Sensaphone Control Center form), select File>Load Programming. Select the programming file you wish to load into the replacement SCADA 3000 unit, then click the Open button. All of the programming information saved in the .s3k file will be transferred to the SCADA 3000 unit.

7) Set the SCADA 3000's internal clock. Click the 'System' button on the extreme left end of the toolbar or select Programming>System from the SCADA 3000 RTU form.

8) Upload C and Ladder programs separately.

Reminder: You are setting the clock INSIDE the SCADA 3000 unit. If you're connected with a phone modem, it's possible that you and the unit are in different time zones.

Why won't my SCADA 3000 dial out?

If you are on-line with the SCADA 3000 through its RS232 port, it will not dial out until you close the connection.

Check that you have a Touch-Tone telephone line. If SCADA 3000 is set for Tone dial, it will not be able to dial out if it's on a Pulse telephone line.

• The Destination telephone number may be incorrectly programmed. Verify the telephone numbers. Is a '1 + area code' prefix required? Or does your telephone line require a '9' prefix to reach an outside line?

• There may be no Call Zone selected. Be sure to check your Call Zone programming for your destinations. Likewise, the Alarm Call Mode may be disabled for this destination. Check and enable the Alarm Call Mode.

• Recognition time may be set too long for an alarm condition to become a valid alarm. You can shorten the Recognition time for the alarm or increase the amount of time that the specific alarm exists. Remember, an unacknowledged alarm exists only when the red alarm LED blinks on the front panel.

• There are no Destinations selected for alarm Notification. For each alarm, you must select which of the 64 dialout destinations to contact.

• 'Max Calling Rounds' might be set to zero, in which case the SCADA 3000 will make no attempt to call. You should increase the calling rounds to a number greater than zero. The default setting is 100.

• The MON switch on the SCADA 3000 unit is in the Disable position.

I keep getting a busy signal when I call my SCADA 3000.

The SCADA 3000 may be trying to deliver an alarm message or a report and the destination intercall delay times are very short. The unit waits the programmed intercall time after contacting a destination before it dials the next number in the sequence. If this period is too short, you'll have only a short time to call in before the SCADA 3000 begins dialing the next destination. Extend the length of the intercall time for each destination.

Why am I getting a temperature reading of -100 degrees F (-75 degrees C)?

You'll get this reading if a temperature sensor has become disconnected or has broken wires. Examine the wires to the temperature sensor and connect or replace the wiring.

Why am I getting a temperature reading of 300 degrees F (150 degrees C)?

• The temperature sensor wires may have shorted or are touching. Examine the temperature sensor wiring and correct or replace the wiring.

• A shunt for setting the input type may not be installed in the TMP/DRY position. Check and move the shunt to the correct position.

Why does SCADA 3000 call again with the same alarm after I just acknowledged it?

• Alarm Recognition time may be set too short. If an alarm is acknowledged, then clears and quickly re-trips, the SCADA 3000 must wait the Alarm Recognition time before calling again. Lengthen the Alarm Recognition time to avoid rapid callbacks for the same alarm.

• The Alarm Reset Time is too short. Alarm Reset Time is the time allowed for an acknowledged alarm's fault condition to be corrected before the unit reactivates the alarm and begins the dialout process all over again.

Why won't the SCADA 3000 main unit communicate with expansion I/O modules?

Every SCADA 3000 expansion I/O module (except the annunciator module) has a built-in communications bus diagnostic feature. If the module's yellow 'PULSE' LED is blinking, communications to that module are working normally. If a module's 'PULSE' LED is off, that module is no longer communicating with the SCADA 3000 main unit. A loss of communication is usually due to one of the following:

• The communications bus is not properly terminated. Termination is required at the extreme ends of the communications network to minimize signal errors. Check that the SCADA 3000 main unit and all expansion modules have their bus termination jumpers in the correct positions. See Chapter 7 (I/O Expansion Modules) for more information on communications Bus Termination.

• The communications bus is not properly wired. The communications bus wiring requires a 4-conductor cable. Two conductors are used for the A&B pair and the other two are used for the Y&Z pair. Make sure the wiring is reversed between the expansion modules and the main unit, but remains straight through between the modules. See Chapter 7 (I/O Expansion Modules) for

more information on communications Bus Termination.

• There is an Expansion Module bus address conflict. Up to 15 I/O expansion modules may be connected to the SCADA 3000 main unit to provide additional inputs and/or outputs. Each module must be configured with its own ***unique*** address using the BUS ADDRESS jumpers. See the user's manual supplement supplied with the I/O expansion module for information on setting BUS addresses.

• There may be a loose or open connection in the communications bus wiring. All 4 wires must be connected to an expansion module for the SCADA 3000 main unit to communicate with it. Check for an open bus connection by using a DC voltmeter and checking the voltage between the Y and Z terminals on the SCADA 3000 MAIN UNIT. If all the bus connections are complete, you will see about 2.4 Volts DC. If you see 4.5 Volts DC, there is an open bus connection somewhere.

Why won't my Ladder-Logic program run?

The CTRL switch on the SCADA 3000 unit is in the Disable position.

Why won't my C program run?

The CTRL switch on the SCADA 3000 unit is in the Disable position.

Appendix I: SCADA 3000 ACCESSORIES

FGD-0006	Magnetic Reed Switch
FGD-0007	Passive Infrared Detector
FGD-0010	Accessory Wire
FGD-0013	Water Sensor
FGD-0022	Temp Alert
FGD-0023	ISOTEL Surge Protector
FGD-0027	Humidistat
FGD-0051	4-Conductor RS-485 Cable
FGD-0052	Humidity Transmitter
FGD-0102	10K Weatherproof Temperature Probe
FGD-0103	10K Indoor Decorator Zone Tempeature Sensor
FGD-0104	10K Outdoor Air Weatherproof Temperature Sensor
FGD-0105	10K Immersion Temperature Sensor
FGD-3010	SCADA 3000 Universal Input Module
FGD-3020	SCADA 3000 Pulse Count Input Module
FGD-3030	SCADA 3000 Thermocouple Input Module
FGD-3040	SCADA 3000 Relay Output Module
FGD-3050	SCADA 3000 Analog Output Module
FGD-3060	SCADA 3000 Annunciator Module
FGD-3100	SCADA 3000 60W Hard-wired Power Supply
FGD-3110	SCADA 3000 55W Plug-in Power Supply
FGD-3200	SCADA 3000 5AH Battery & Mounting Kit
FGD-3210	SCADA 3000 18AH Battery & Mounting Kit
FGD-3400	SCADA 3000 33.6 Modem Option
FGD-3500	SCADA 3000 Voice Module Option

Appendix J: SCADA 3000 REPLACEMENT PARTS

BAT-0004	CR2430 3V Lithium Battery
BAT-0012	5.2AH 12V Sealed Gel-Cell Battery
BAT-0013	18AH 12V Sealed Gel-Cell Battery
CON-0027	16 Position Terminal Block Plug
CON-0033	2 Position Jumper Shunt
CON-0034	4 Position Terminal Block Plug
CON-0108	4 Position Jumper Shunt
CON-0120	9 Position Terminal Block Plug
FUS-0005	500mA TR5-style Time-Lag fuse (Wickman #3720500041)
FUS-0006	4 A 250V, size TR-5 Fuse (Wickmann #3721400041)
HSG-0013	SCADA 3000 Enclosure Base
HSG-0014	SCADA 3000 Enclosure Cover
LIT-0020	SCADA 3000 Instruction Manual

Appendix K: RETURNING THE UNIT FOR REPAIR

In the event that the SCADA 3000 does not function properly, we suggest that you do the following:

1) Record your observations regarding the SCADA 3000 malfunction.

2) Call the Technical Service Department at (610)558-2700 prior to sending the unit to Phonetics for repair.

If the unit must be sent to Phonetics for Servicing, please do the following:

1) Turn the power switch Off, disconnect all wiring and unplug the unit.

2) Carefully pack the unit to avoid damage in transit. Use the original container (if available) or a sturdy shipping box.

3) You must include the following information to avoid shipping delays:

   a) Your name, address and telephone number.

   b) A note explaining the problem.

4) Ship your package to the address below:

   SERVICE DEPARTMENT

   Phonetics Inc.

   901 Tryens Road

   Aston, PA 19014

5) Ship prepaid and insured via UPS or US Mail to ensure a traceable shipment with recourse for damage or replacement.

Important Information for Canadian Customers

In the event that your Sensaphone SCADA 3000 unit does not function properly, Canadian customers have the option of shipping the unit to one of the following Phonetics-authorized Canadian Repair facilities:

Microwise Computer Systems
100 Covington Crescent
Kitchener, Ontario N2N 2X3
(519) 744-9892

Galvanic Analytical Systems, Ltd.
5664 Burleigh Crescent S.E.
Calgary, Alberta T2H 1Z8
(403) 253-6576

Please record your observations regarding the unit's malfunction and follow the procedures outlined on the previous page.

For Technical Support questions, you may call Phonetics Technical Service Department at (610) 558-2700, or by E-mail at support@sensaphone.com.

Appendix L: PROGRAMMING FORMS

I. CONFIGURATION

Type (Serial/Phone/Radio):_____

Baud Rate:_____

Unit Description(Name):_____

Phone #:_____

Comm Port:_____

Slave ID:_____

Dial Method(tone/pulse):_____

Dial Pr efix (if any):_____

Alphanumeric Pager Speed:_____

Voice Repetitions per call:_____

Rings T ill Answer: _____

II. INPUTS PROGRAMMING (0-15)

Input types: Closure, Run Time, 0-5 volt, 4-20mA, 10K deg. F, 10K deg C

Unit choices: None, Open/Closed, Closed/Open, On/Off, Off/On, Running/Stopped, Stopped/Running, Deg. F, Deg. C, Inches, Feet, Gallons, Liters, PSI, GPM, Volts, Amps, Hours, Minutes, Seconds, Custom

#	Name	Type	Units	Table Low	Table High	Calibration Factor
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

II. OUTPUTS PROGRAMMING (0-7)

Output Types: Relay

Unit choices: None, Open/Closed, Closed/Open, On/Off, Off/On, Running/Stopped, Stopped/Running, Deg. F, Deg. C, Inches, Feet, Gallons, Liters, PSI, GPM, Volts, Amps, Hours, Minutes, Seconds, Custom

#	Name	Type	Units	Table Low	Table High	Offset	Forced Status(Not/Off/On)
0							
1							
2							
3							
4							
5							
6							
7							

IV. ALARMS PROGRAMMING (0-31)

Alarm #	Source UAF	Alarm Name	Custom Message #	Dialout Destination Numbers	Type (N.O./ N.C./Limit)	Recog Time	Reset Time	Limit Low	Limit High	Enable (Y/N)
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										

IV. ALARMS PROGRAMMING (32-63)

Alarm #	Source UAF	Alarm Name	Custom Message #	Dialout Destination Numbers	Type (N.O./ N.C./Limit)	Recog Time	Reset Time	Limit Low	Limit High	Enable (Y/N)
32										
33										
34										
35										
36										
37										
38										
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51										
52										
53										
54										
55										
56										
57										
58										
59										
60										
61										
62										
63										

V. DIALOUT DESTINATIONS (0-31)

Alarm Call Modes: Disabled, Until Acknowledged, Inform Dial Types: Fax, Voice, E-Mail, Pager, Alpha Pager, Modem

Name	Number/Address	Call Zone(s)(1-4)	Alarm Mode	Dial Type	Call Delay	Report(Y/N)
0						
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						

V. DIALOUT DESTINATIONS (32-63)

Name	Number/Address	Call Zone(s)(1-4)	Alarm Mode	Dial Type	Intercall Delay	Report (Y/N)
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						

VI. CALL ZONE PROGRAMMING

The 24 hour day can be split into 3 time periods that can be used throughout the week. The day begins at 12:00AM and a programmable morning and evening time can be entered. For example, the day period may begin at 8:00AM, and the evening period may begin at 6:00PM. In this scenario, the three time periods would be:

Period 1: 12:00AM to 8:00AM
Period 2: 8:00AM to 6:00PM
Period 3: 6:00PM to 12:00AM

Since there are 7 days per week and three time periods, we have now identified 21 time periods over the course of the week. The SCADA 3000 allows you to apply a zone number to each of the 21 time periods. Up to four zones may be defined. For example, you could define Monday through Friday from 8:00AM until 6:00PM as Zone 1, Saturday and Sunday from 8:00AM to 6:00PM as Zone 2, and so forth. An example follows. Fill in the bottom diagram to configure your own call zones.

	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
12:00AM	2	3	3	3	3	3	2
Morning: 8:00AM	2	1	1	1	1	1	2
Evening: 6:00PM	2	4	4	4	4	4	2

	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
12:00AM							
Morning:_____							
Evening:_____							

VII. STATUS REPORT & CUSTOM VOICE MESSAGES

GENERAL SETTINGS:

Number of Custom Phrases ☐ 32 ☐ 64

☐ Use Touch-Tone Password for Incoming Calls

☐ Use Password for Local Voice Port

☐ Recite ID Phone Number in Status Report

☐ Recite Custom ID Message in Status Report

☐ Speak Canned Messages with Custom Messages

STATUS REPORT INCLUDES:

Recite Alarms: ☐ None ☐ All ☐ Existing Alarms

☐ Give Main Power Status

☐ Give Main Power Supply Consumption

☐ Give Status Messages

☐ Recite Battery Voltage

☐ Give Ladder Program Status

☐ Give C Program Status

VII. CUSTOM VOICE MESSAGES

#	Description	Status UAF	Alarm #	#	Description	Status UAF	Alarm #
0				32			
1				33			
2				34			
3				35			
4				36			
5				37			
6				38			
7				39			
8				40			
9				41			
10				42			
11				43			
12				44			
13				45			
14				46			
15				47			
16				48			
17				49			
18				50			
19				51			
20				52			
21				53			
22				54			
23				55			
24				56			
25				57			
26				58			
27				59			
28				60			
29				61			
30				62			
31				63			

VIII. DATA LOG (up to 128 points)

Interval: [ ] Hours [ ] Minutes [ ] Seconds

Available UAFs: Inputs, Outputs, Timers, Counters, Bits, Ladder Variables, C Variables, Gas Flow, Alarms, Diagnostics

Selected UAFs	UAFs	UAFs	UAFs	UAFs	UAFs	UAFs	UAFs

IX. REPORTS

Event Log (Y/N)	# of Records	Data Log (Y/N)	# of Records	Report Interval
				(hrs.) : (mins):

X. LCD DISPLAY

Available UAFs for display: Inputs, Outputs, Timers, Counters, Bits, Ladder Variables, C Variables, Gas, Alarms, Diagnostics

Selected UAFs	UAFs	UAFs	UAFs	UAFs	UAFs	UAFs	UAFs

XI: PID CALCULATIONS

For each PID algorithm, fill in the necessary information. Note: Each parameter MUST have a value entered for that algorithm to calculate correctly.

	PID 0	PID 1	PID2	PID 3	PID 4	PID 5	PID 6	PID 7
Name								
Input UAF								
Output UAF								
Reload Time								
Target Set Point								
Dead Zone								
Tuning Factors:								
Kp Factor/UAF								
Ki Factor/UAF								
Kd Factor/UAF								
Use Output Limiting (Y/N)								
Min:								
Max:								
Enable PID (Y/N)								

XII: GAS FLOW CALCULATIONS

For each Meter run (flow), fill in the appropriate information.

General Setup:	
Standard Pressure (PSIA)	
Standard Temperature (DEG F)	
Contract Base Pressure (PSIA)	
Contract Base Temperature (DEG F)	
Flow Start Time	

CHANNEL SETUP: Name	**FLOW 0**	**FLOW 1**	**FLOW 2**	**FLOW 3**
Type (Orifice Plate/Displacement)				
Orifice Plate Type (Steel/Monel/Carbon Steel)				
Orifice Plate Bore Diameter (in.)				
Bore Diameter Reference Temperature (Deg. F)				
Meter Tube Type (Steel/Monel/Carbon Steel)				
Meter Tube Internal Diameter (in.)				
Pulses Per Cubic Foot (Positive Displacement only)				
Specific Gravity				
Carbon Dioxide Content (mole %)				
Nitrogen Content (mole %)				
Dynamic Viscosity (lbs-mass per foot-sec)				
BTU Factor (BTUs/ft^3)				
Low Flow Cutoff (inches of H_2O)				
Static Pressure UAF				
Differential Pressure UAF				
Temperature UAF				
Pulses UAF (Positive Displacement only)				

Index

A

Alarm Acknowledgment **11:** 17–23
  Alarm Dialing Priority **11:** 20
  Alphanumeric Pager Dialout **11:** 19–20
  alphanumeric pager dialout **11:** 13
  Computer Alarm Reception **11:** 20–23
    Alarm Printing and Logging **11:** 21
    Audible Alarm **11:** 22–23
    Logging **11:** 22
    Printing **11:** 22
  Maximum Calling Rounds **11:** 20
  Modem Dialout **11:** 20
  Pager Dialout **11:** 18–19
  Viewing Current Alarms **11:** 20
  Voice Dialout **11:** 17–18
Alarm Dialout **11:** 16–17
(see also PC Alarm Dialout)
  Alphanumeric Pager **11:** 17
  E-Mail **11:** 17
  Fax **11:** 17
  Modem **11:** 17
  Numeric Pager **11:** 17
  Voice **11:** 16
Alarm Notification **11:** 15–17
  Alarm Call Modes **11:** 16–17
    Inform **11:** 16
    Until Acknowledged **11:** 16
  Alarm Recognition **11:** 16
  Notification **11:** 16
Alarm Programming **11:** 12–15
(see also PC Alarm Dialout)
  I/O Point Value **11:** 13
  State **11:** 13
  Status **11:** 13
Alarm Setup **11:** 13–15

C

C Program **16**
  Specifications **16:** 1
C Programming **16**
  Arrays **16:** 10
  Differences between SCADA & Standard **16:** 16–21
  Editor **16:** 12–15
    Start & Stop **16:** 15
    Uploading, Compiling, Starting **16:** 13
  Error Handling **16:** 12–15
  Function Library **16:** 7–11
  Keywords **16:** 4–6
  Language & Structure **16:** 1–6
  Predefined Variables **16:** 6
  Sample Program **16:** 10
  Structure **16:** 2–6
  Synchronous vs. Asynchronous **16:** 15
  Viewing the Variables **16:** 16
Carrier Wait Time **10:** 5
Communicating with SCADA 3000 **2:** 2, **9, 12:** 6
  Creating New Units **2:** 2, **9:** 3
  Communications Programming **9:** 1
  Radio vs. Phone **9:** 7
  RS232 Ports **9:** 7-8, **E:** 1–2
    Pin Description & Data Direction **E:** 1–2
  Running the Software **9:** 1

D

Datalogging **11:** 23–24
  Capacity **11:** 24
  Downloading the Data Logger **11:** 26
  Order of Data Points Logged **11:** 25
  Resetting **11:** 26
  Selecting Data Points to Log **11:** 24
  Start Time **11:** 25
  Stopping the Data Logger **11:** 26
  Time Interval **11:** 25
  Viewing the Data **11:** 26
    Graph **11:** 27
Destinations Programming **11**
  Alarm Call Mode **11:** 4-5
    Disabled **11:** 4
    Inform **11:** 4–5
    Until Acknowledged **11:** 4
  Alphanumeric Pagers **11:** 1–3
  Call Zones **11:** 3
  Destination **11:** 1
  Dial Type **11:** 5
  E-Mail **11:** 2
  Intercall Delay **11:** 5

Name **11:** 1
Numeric Pagers **11:** 1
Send Report **11:** 5
Special Dialing Codes **11:** 2
Diagnostics **19**
Main Board **19:** 1
Module Communications **19:** 2
Distributed Control Using Visual Basic **23**
About Visual Basic **23**: 6
Definitions **23**: 7–9
Dim Statement **23**: 7
Do...Loop Statement **23**: 8
For...Next Statement **23**: 8
If...Then...Else Statement **23**: 7
Keywords **23**: 7
Operators **23**: 9–10
Assignment **23**: 9
Comparators **23**: 9
Month, Day, Year, Hours... **23**: 10
Programming Methodology **23**: 10–11
Properties of the Script Component **23**: 5
Referencing I/O Points **23**: 3–5
Running your Program **23**: 5
Sample Application **23**: 3
Saving your Program **23**: 5
Syntax Checking **23**: 4
Requirements **23**: 1
Sample Program **23**: 11
Script Editor **23**: 1–3
Structure **23**: 6–7

E

E-mail Server Option **21**
Event Logger **11**: 28
Deleting Events **11**: 29
Downloading **11**: 28
Printing, Exporting, & Copying **11**: 29
Viewing **11**: 28–29

F

Firmware **Appendix F**
Updating **F:** 1–2

G

Gas Flow **18:** 1–6
Calculated Information **18:** 5–6
Orifice Plate Method **18:** 1–3
Positive Displacement Method **18:** 1
Programming **18:** 2–3

I

I/O Expansion Modules **7:** 1–4
Communications Bus Wiring **7:** 2–3
Power Supply **7:** 2
Types **7:** 4
I/O Programming **10:** 11–15
Input Type **10:** 12
Outputs **10:** 14–15
UAF **10:** 13-15
Tables and Calibration **10:** 13
Units **10:** 12
Input Configuration **5**
10k Thermistors **5:** 3–6
Curve Data **Appendix D**
4-20mA Current Loop Transducers **5:** 5
Analog Voltage Output Transducers **5:** 4–6
Correct Jumper Positioning **5:** 2
Normally Open/Normally Closed **5:** 2
Internet Web Status **20**
Formatting the Web Page **20:** 5
Browser Refresh **20:** 5
Logo Settings **20:** 5
Frequently Asked Questions **20:** 7–8
Global Settings **20:** 2–4
Global Status **20:** 2
.wml format **20:** 3
Internet Settings **20:** 1–2
Requirements **20:** 1
Web Status Options **20:** 6–7
Alarm Options **20:** 6
FTP Delivery Options **20:** 6–7
Instant Web Page **20:** 7
Local Web Pages & Filenames **20:** 6
Online Options **20:** 6
Web Status Polling **20:** 4-7
Selecting Data Points **20:** 4–5
Setting up a Polling Schedule 20: 7

K

Keypad Programming **13**
Navigating the Display by **13:** 2
Using the Keypad **13:** 3–5

L

Ladder Logic **15**
- Comparison Instructions **15:** 11–14
- Compiling, Verifying, Uploading **15:** 21
- Deleting Rungs/Instructions **15:** 21
- Elements & Functions **15:** 2
- Entering UAFs, Titles, Comments **15:** 20
- Examples **15:** 27–30
- Ladder Run Viewer **15:** 23–30
- Math Instructions **15:** 14–17
- Performance times **15:** 26–29
- Printing **15:** 23
- Program Development **15:** 17–18
- Rules & Conventions of **15:** 2
- Running/Stopping Execution **15:** 22–30
- Saving/Loading Files **15:** 21
- Statistics **15:** 23
- Starting a New Ladder Program **15:** 18-23
  - Adding Rungs **15:** 19
  - Branching **15:** 19
  - Changing Instructions **15:** 20–30
  - Cut, Copy, Paste **15:** 21–30
- Supported Elements & Instructions **15:** 1
- Syntax Checking **15:** 21–30
- Viewing Ladder Variables **15:** 23-26
  - Bits **15:** 25–26
  - Counters **15:** 25
  - Timers **15:** 24–25
  - Variables **15:** 24

LCD Programming **13**
- Alarms **13:** 5–6
- Grouping Data Points **13:** 2
- Keypad Use **13:** 3-6
  - Data Points **13:** 3
  - Return **13:** 6
  - System Status **13:** 6
- Programming Data Points for Scroll Mode **13:** 1
- Removing Data Points **13:** 2
- Scrolling Display **13:** 2–3

M

Modem Card Installation **3:** 4–5

O

Output Wiring **6**

P

Pagers, Programming Destinations for **11**
- Special Alphanumeric Dialing Codes **11:** 3

PC Alarm Dialout **22**
- Alarm Programming **22:** 6
  - Alarm Acknowledgement **22:** 6
- Computer Requirements **22:** 1
- Programming Destinations **22:** 5–6
- Sample Alarm Dialout Sequence **22:** 6
  - Alphanumeric Pager **22:** 7
  - Fax & Email **22:** 7
  - Numeric Pager **22:** 7
  - Voice **22:** 6–7
- SCADA 3000 Programming for **22:** 3–5
  - Voice Messages **22:** 4–5
- Software Setup **22:** 1–3
  - Email Settings **22:** 2–3
  - Modem Commands **22:** 2
  - Telephone Voice Volume **22:** 3

Phone Line Applications, Programming for **11**
- Destinations **11:** 1–5

PID Programming **17:** 1–4
- PID Algorithm **17:** 2–4
- Tuning **17:** 3–4

Polling **11:** 29 *(see also Internet Web Status Polling)*
- Changing the Schedule **11:** 31
- Setting Up a Schedule **11:** 29–31

R

Radio Modem Applications **12**
- Communicating with SCADAs **12:** 6
- Configuring SCADA 3000 for **12:** 1–2
- Flow Control **12:** 2
- Host Computer Setup **12:** 2–5
- Selecting Radio Modems **12:** 1
- Troubleshooting **12:** 6–7

Real-Time Output Control **14:** 31

Real-Time Screen Design **14**
- AudVid Tab **14:** 27–29
- Contain Tab **14:** 21–23
- Datapoint Tab **14:** 13–17
  - Properties for Datapoint Alert **14:** 16
  - Properties for DataPointShape **14:** 16
  - Properties for Labels/Values **14:** 14–15
- Dynamic Tab **14:** 29–31
- Editing **14:** 1–3
- Electrical Tab **14:** 17–19

Gauge Tab **14:** 25–27
Setting Up Gauges **14:** 25
Setting up Strip Charts **14:** 25
Graphic Tab **14:** 10–13
Group Tab **14:** 7–10
Loading for Editing **14:** 32
Palette **14:** 1–7
Advanced Properties **14:** 5–7
Basic Properties **14:** 3–5
Placing Objects on **14:** 4–7
Right-click tools **14:** 6–7
Sizing **14:** 1
Rotate Tab **14:** 19–21
Running **14:** 32
Saving **14:** 31–32
Selecting SCADA Units for **14:** 3
Valve Tab **14:** 23–25
Reports **11:** 22-23
Rings Until Answer feature **10:** 5

S

SCADA 3000
Backup Battery **4:** 4
Estimating Power Consumption **4:** 2–4
Firmware, Updating **Appendix F**
General Description **1:** 2–6
General Wiring Considerations **5:** 1
Grounding & Power Surge Protection **4:** 2–4
Mounting Instructions **3**
Operating Environment **3:** 1
Power Requirements **4:** 1
Power Supply Fuse **4:** 2
Software Installation **8**
from CD **8:** 2
from Diskettes **8:** 2
Internet Updates **8:** 2
Minimum Requirements **8:** 1
Technical Support **1:** 5
Software Internet Upgrade **8:** 2**, Appendix G**
Solar Cell Powering **4:** 5
System Programming **10**
Clock **10:** 3
Auto Daylight Savings **10:** 3
Unit Date & Time **10:** 3
Dialout Settings **10:** 4–15
Alpha Pager Speed **10:** 4
Dialing Method **10:** 4
Dialing Prefix **10:** 4
Maximum Calling Rounds **10:** 4
Online Timeout **10:** 5
Voice Repetitions **10:** 4
Slave Address **10:** 3
Unit ID Number **10:** 3
Unit Name **10:** 3
System Security **9:** 4–8
Changing Passwords **9:** 6–7
Setting Up User Accounts **9:** 6–8
User Log-in with Security **9:** 6–8

T

Terminal Blocks **5**
Removing **5:** 1–2

U

UAFs (Universal Address Format) **10:** 5–10
Specifying & Applying **10:** 6
Alarms **10:** 9
Bit **10:** 7–8
C Variable **10:** 8
Counter **10:** 7
Diagnostics **10:** 9–15
Input **10:** 6
Ladder Variable **10:** 8
Output **10:** 6
User Inactivity Timer **10:** 8

V

Voice Board Installation **3:** 6–7
Voice Programming **11:** 6–12
Alarm Status Types **11:** 7–8
Linking Messages to UAFs **11:** 9
Local **11:** 10
Messages **11:** 6-8
Custom **11:** 6–7
Password for Incoming **11:** 7
Password for Local **11:** 7
Recite ID Phone Number **11:** 7
Recording & Programming **11:** 6–31
Recording and Playback **11:** 10
Remote **11:** 11
Status Report-Customizing **11:** 8
To Hear a Status Report **11:** 12

3 YEAR LIMITED WARRANTY

1. **WARRANTOR:** Dealer, Distributor, Manufacturer

2. **ELEMENTS OF WARRANTY:** This Product is warranted to be free from defects in materials and craftsmanship with only the limitations and exclusions set out below

3. **WARRANTY AND REMEDY: Three-Year Warranty** — In the event that the Product does not conform to this warranty at any time during the time of three years from original purchase, warrantor will repair the defect and return it to you at no charge

This warranty shall terminate and be of no further effect at the time the Product is (1) damaged by extraneous cause such as fire, water, lightning, etc. or not maintained as reasonable and necessary; (2) modified; (3) improperly installed; (4) repaired by someone other than warrantor; (5) used in a manner or purpose for which the Product was not intended; or (6) sold by original purchaser.

WARRANTORS' OBLIGATION UNDER THIS WARRANTY IS LIMITED TO REPAIR OR REPLACEMENT OF THE PRODUCT. THIS WARRANTY DOES NOT COVER PAYMENT OR PROVIDE FOR THE REIMBURSEMENT OF PAYMENT OF INCIDENTAL OR CONSEQUENTIAL DAMAGES.

It must be clear that the warrantors are not insuring your premises or guaranteeing that there will not be damage to your person or property if you use this Product. The warrantors shall not be liable under any circumstances for damage to your person or property or some other person or that person's property by reason of the sale of this product or its failure to operate in the manner in which it is designed. The warrantors' liability, if any, shall be limited to the original cost of the Product. The warrantors assume no liability for installation of the Product and/or interruptions of the service due to strikes, riots, floods, fire, and/or any cause beyond Sellers' control.

4. **PROCEDURE FOR OBTAINING PERFORMANCE OF WARRANTY:** In the event that the Product does not conform to this warranty, the Product should be shipped or delivered freight prepaid to a warrantor with evidence of original purchase.

5. **LEGAL REMEDIES:** This warranty gives you specific legal rights, and you may also have other rights which vary from state to state to the extent allowed by law expressly in lieu of any other express or implied warranty, condition, or guarantee.

Effective date: 1 June 1999

Phonetics, Inc.

901 Tryens Road

Aston, PA 19014

Phone: (610) 558-2700 Fax: (610) 558-0222